Phylogenetic Assembly of Paleogenomes Integrating Ancient DNA Data

Nina Luhmann

Phylogenetic Assembly of Paleogenomes Integrating Ancient DNA Data

Ph. D. Thesis

submitted to the
Faculty of Technology,
Bielefeld University, Germany
for the degree of Dr. rer. nat.

by

Nina Luhmann

March, 2017

Referees:

Prof. Dr. Jens Stoye
Prof. Dr. Cedric Chauve
Prof. Dr. Annie Chateau

Gedruckt auf alterungsbeständigem Papier nach DIN-ISO 9706.
Printed on non-aging paper according to DIN-ISO 9706.

Zusammenfassung

In der komparativen Genomik ist die Rekonstruktion der Genome anzestraler Spezies ein wichtiges Problem, um deren Evolution analysieren zu können. Die Diversität heutiger Genome in Bezug auf Mutationen und Umordnungen der Genomsequenz erlaubt es, die Dynamik der evolutionären Prozesse aufzudecken, die zur Entwicklung heutiger Spezies ausgehend von einem gemeinsamen Vorfahren geführt haben. Diese Artenbildung wird in einem phylogenetischen Baum abgebildet. Komparative Methoden zur Rekonstruktion anzestraler Genome zielen darauf ab, genomische Merkmale wie die Reihenfolge von Markern (z.B. Gene) für bereits ausgestorbene Spezies an den internen Knoten des Baums unter verschiedenen evolutionären Modellen abzuleiten. Dabei stützen sich diese Methoden lediglich auf die vorhandenen Informationen für rezente Genome an den Blättern des phylogenetischen Baums.

In der letzten Zeit hat der stetige Fortschritt in der Sequenziertechnologie das Feld der Paleogenomik geprägt, in welcher sich Studien vorzeitlicher DNA (sogenannter *ancient DNA* (aDNA)) aus konserviertem organischem Material zusehends mit der Sequenzierung und Analyse ganzer Paleogenome beschäftigen. Solche „genetischen Zeitreisen“ erlauben direkte Einblicke in spezifische Phasen der Evolution, welche nicht nur implizit von rezenten DNA-Sequenzen abgeleitet sind. Da DNA jedoch nach dem Tod eines Organismus auf natürliche Weise abgebaut wird und Umweltbedingungen die Konservierung der DNA beeinflussen, können meist nur sehr kurze DNA-Fragmente sequenziert werden. Dies verhindert eine detaillierte Analyse von Umordnungen in den Genomen entlang der Kanten des phylogenetischen Baums.

Das Ziel dieser Arbeit ist die Kombination von aDNA-Daten und komparativer Rekonstruktion anzestraler Genome im phylogenetischen Kontext. Der Vergleich von rezenten verwandten Genomen kann dabei helfen, eine Anordnung für aDNA-Fragmente abzuleiten, während die aDNA-Sequenzdaten als zusätzliche Informationsquelle

in komparativen Rekonstruktionsmethoden einbezogen werden können, um die Rekonstruktion aller Vorfahren im phylogenetischen Baum zu verbessern. Unser erster Fokus liegt auf integrativen Methoden, welche Genanordnungen unter der Annahme von Parsimonie global in der Phylogenie rekonstruieren. Ein zu Grunde liegendes Distanzmodell für Umordnungen beschreibt dabei die evolutionären Operationen, welche entlang der Kanten des phylogenetischen Baums aufgetreten sein können. Während komplexe Modelle Einsicht in die biologischen Mechanismen der Evolution geben können, ist das Problem der anzestralen Rekonstruktion mit diesen Distanzmodellen aus informatischer Sicht jedoch NP-schwer. Eine Ausnahme ist die sogenannte *Single-Cut-or-Join (SCJ)*-Distanz, welche eine auf Markerordnungen basierende Repräsentation der involvierten Genome nutzt, um einfache Brüche und Verknüpfungen in der Anordnung von Markern zu modellieren.

Wir beschreiben Rekonstruktionsmethoden mit dem Ziel, die SCJ-Distanz im Baum zu minimieren. Zusätzlich setzen wir voraus, dass die rekonstruierten Lösungen konsistent sind, d.h. sie repräsentieren lineare oder zirkuläre Regionen eines Genoms. Unsere erste Methode hat eine polynomielle Laufzeitkomplexität und basiert auf dem Sankoff-Rousseau-Algorithmus. Die Methode integriert explizit aDNA-Fragmente und mögliche Verknüfungen an einem inneren Knoten des Baumes. Wir zeigen, dass der Einbezug von Kantenlängen im Baum in der Praxis eine eindeutige optimale Lösung ergibt. Unser zweiter Ansatz verfolgt eine allgemeinere Strategie, indem neben der SCJ-Distanz die aDNA-Sequenzierdaten als lokale Gewichte für Markernachbarschaften in der Zielfunktion einbezogen werden. Wir beschreiben einen parametrisierten Algorithmus für dieses Problem, welcher auch die Berechnung aller optimaler Lösungen erlaubt. Zuletzt beschreiben wir einen Ansatz, um die Lücken in Markeranordnungen mit Hilfe von aDNA-Daten zu schließen und so vollständige Paleogenomsequenzen zu rekonstruieren, unterstützt durch verwandte rezente Genomsequenzen. Dies erlaubt uns außerdem, widersprüchliche Markeranordnungen auf Grundlage der Sequenzdaten aufzulösen.

Wir evaluieren unsere Modelle und Algorithmen mit simulierten und biologischen Daten. Wir konzentrieren uns besonders auf zwei aDNA-Sequenzdatensätze des Krankheitserregers *Yersinia pestis*, welcher als Ursache mehrerer Pandemien im Mittelalter gilt. Wir zeigen, dass die Kombination von aDNA-Sequenzdaten und die Rekonstruktion im phylogenetischen Baum zu einer deutlich reduzierten Fragmentierung der aDNA-Daten führt und können mit Hilfe der vielfältigen Methoden auch alternative Rekonstruktionen vergleichen, um zuverlässig rekonstruierte Regionen zu unterstreichen.

Abstract

In comparative genomics, reconstructing the genomes of ancestral species in a given phylogeny is an important problem in order to analyze genome evolution over time. The diversity of present-day genomes in terms of local mutations and genome rearrangements allows to shed light on the dynamics of evolutionary processes that led from a common ancestor to a set of extant genomes. This speciation history is depicted in a phylogenetic tree. Comparative genome reconstruction methods aim to infer genomic features such as an order of markers (e. g. genes) for extinct species at internal nodes of the tree by applying different evolutionary models, relying only on the information available for the extant genomes at the leaves of the phylogenetic tree.

Recently, the steady progress in sequencing technologies led to the emergence of the field of paleogenomics, where the study of ancient DNA (aDNA) found in conserved organic material is moving rapidly towards the sequencing and analysis of complete paleogenomes. Such "genetic time travel" allows direct insight into specific phases of the evolution of specific genomes that are not only implicitly inferred from extant DNA sequences. However, as DNA is naturally degraded over time after the death of an organism and environmental conditions interfere with the conservation of DNA material, an assembly of these paleogenomes is usually fragmented, preventing a detailed analysis of genome rearrangements along the branches of the phylogenetic tree.

In this thesis, we aim to combine the study of aDNA and comparative ancestral reconstruction in a phylogenetic framework. The comparison with extant related genomes can naturally assist in scaffolding a fragmented aDNA assembly, while the aDNA sequencing data can be included as an additional source of information for comparative reconstruction methods to improve the reconstructions of all related genomes in the phylogenetic tree. Our first focus is on integrative methods to reconstruct marker orders globally in a phylogeny under the assumption of parsimony. An un-

derlying rearrangement model can describe the evolutionary operations that occurred along the edges of the tree. However, as much as complex rearrangement scenarios can give insights into underlying biological mechanisms during evolution, from an computational point of view the ancestral reconstruction problem under rearrangement distances is an NP-hard problem. One exception is the Single-Cut-or-Join (SCJ) distance, that uses a marker order-based representation of the involved genomes to model the cut and join of marker adjacencies as evolutionary operations.

We build upon this rearrangement model and describe parsimony-based reconstruction methods aiming to minimize the SCJ distance in the tree. In addition, we require the reconstructed solutions to be consistent, such that they represent linear or circular regions of the ancestral genome. Our first polynomial-time method is based on the Sankoff-Rousseau algorithm and directly includes an aDNA assembly graph at one internal node of the tree. We show that including branch lengths in the underlying tree can avoid ambiguity in practice. Our second approach follows a more general strategy and includes the aDNA sequencing data as local weights for adjacencies next to the SCJ distance in the objective. We describe a fixed-parameter-tractable algorithm that also allows to sample co-optimal solutions. Finally, we describe an approach to fill gaps between potentially adjacent markers by aDNA data to reconstruct the complete genome sequence of a paleogenome guided by the related extant genome sequences. In addition, this approach enables us to select the adjacencies that are supported by the sequencing information from sets of conflicting adjacencies.

We evaluate our proposed models and algorithms on simulated and biological data. In particular, we integrate two aDNA sequencing data sets for ancient strains of the pathogen *Yersinia pestis*, that is understood to be the cause of several pandemics in medieval times. We show that the combination of aDNA sequencing reads and a parsimonious reconstruction in the phylogenetic tree reduces the fragmentation of an initial aDNA assembly substantially and explore alternative reconstructions to emphasize reliably reconstructed regions of the ancient genomes.

Acknowledgments

In the past years, I was enabled to enjoy the privilege of a scientific education and would like to thank everyone who directly or indirectly made this work possible.

First of all, I would like to thank Roland Wittler for all kinds of support in the last years, from basically reading everything I ever wrote to giving advice when needed and especially organizing the best graduate school I know! I am thankful to Jens Stoye for keeping me attached to the GI group ever since my first teaching assistant job years ago and finally accepting me as a PhD student. A lot of the work in this thesis started during my visit in Vancouver working with Cedric Chauve. I am very grateful for all the time he invested, the nearly infinite number of research ideas and the contagious enthusiasm that all together taught me a lot besides research as well, merci! Thank you also to Annie Chateau for accepting to review this thesis.

A big thank you goes to the population of U/V-10 for a great atmosphere to work in and all the important things next to it! Especially thank you to Guillaume for being an awesome office mate, for ping pong breaks and french translations. Thank you to Linda for being a good friend from day one and all the fun we had in the last years! I am grateful to everyone who took some time to read and correct this thesis.

Beyond the time of this PhD, I am thankful for all friends who made me feel at home in Bielefeld. I was lucky to meet most of them at my very first day of university: Jenni, Kathrin, Linda again, Maureen, Farrin, Kai, Flo, Dani...you are to blame that I did not drop out after the very first semester! Now see what you have done.

A big hug goes to my family for always supporting me in all these years, for rarely asking when I will finally be finished and even being interested in what exactly I am working on. Especially thank you to my sister for accepting my weirdness and simply always being there. Everyone needs a house to live in, but a supportive family is what builds a home.

Finally, I would like to acknowledge funding from the International DFG Research Training Group “Computational Methods for the Analysis of the Diversity and Dynamics of Genomes” GRK 1906/1 for the last three years.

Contents

Chapter 1

Introduction

The genetic information of all living organisms is stored in their *DNA* that is composed of smaller units called *nucleotides*. Each cell contains a copy of the genetic information, where *genes* among other things serve as the instructions to build proteins that provide various functionality to the organism. In eukaryotes, DNA is found in the cell nucleus as well as mitochondria and chloroplasts, while in prokaryotes the DNA is contained in the nucleoid in the cytoplasm. The *genome* of an organism denotes its complete set of DNA organized into linear or circular *chromosomes*.

Ever since Darwin described the idea of evolution in the tree of life where all species descended from common ancestors over time [37], the dynamics of genome evolution through mutations has been studied. Understanding the evolutionary processes underlying the development of present-day-species is a key goal of evolutionary genomics. Mutations can be local modifications of the DNA like *substitutions*, *insertions* or *deletions* that directly influence the transcription and translation of genes, e. g. into long chains of *amino acids*. Other evolutionary processes describe the rearrangements of genomic sequences through larger operations such as *inversions* or *translocations* for example. Following the principle of natural selection, some of these modifications get fixed in a population of organisms over time as they are inherited between different generations.

Nowadays, DNA sequencing technologies make it possible to determine the precise order of nucleotides within a DNA molecule. However only relatively short sequences of the DNA strands can be read at the same time by the sequencing machines, resulting in so called *reads*. The lengths of the reads produced by so called *next-generation methods* is typically not longer than around 300 bp, while emerging long-read technologies are able to extend these read lengths at an expense of sequencing accuracy [56]. The

reads then ideally cover the whole genome in an overlapping fashion. The subsequent problem of genome *assembly* describes the process of reconstructing the complete genome sequence from the sequencing reads. It is widely studied in bioinformatics, as the nuleotide sequence of a complete genome is a first step to unravel the genetic information it contains. However it is also one of the most complex computational problems, as regions of the genome can be repeated multiple times and hence do not allow to solve the assembly problem unambiguously [99, 132].

The field of *comparative genomics* describes the comparison of sequenced genomes to study evolutionary processes. It includes the analysis of specific evolutionary operations, gene functions or the analysis of cancer genome evolution. The central problem this thesis concentrates on is the reconstruction of whole ancestral genomes through the analysis of the order of conserved regions. Comparing the genomes of different species completely or partially provides the opportunity to analyze the dynamics of genome evolution through the diversity observed in these extant genomes. For example in the case of human pathogens, such analysis is a key to understand the emergence of pathogenicity or the development of virulence. Even though related extant genomes provide only indirect insight into the true genome structure of extinct species, comparing their genomes allows to infer likely genomic features of ancient species under appropriate models of evolution to ensure the biological plausibility of the reconstructions. Different approaches have been applied to propose reconstructions for a variety of ancestral genomes of plants [39, 94, 97], mammals [16, 29, 84, 154], vertebrates [12, 100, 106], insects [101], or yeast [27], to name a few examples.

In the next section, we will discuss the problem of ancestral genome reconstruction and provide some background for models of genomes and their evolution, before introducing different methods that aim to solve this problem. We will complement this by elucidating new sources of input data, which can improve the reconstruction of ancestral genomes.

1.1 Background

The term *paleogenomics* [15] describes the study of ancient genomes by recovering and understanding the genomic information in long extinct species. On the one hand, the term can be seen in the context of mostly computational comparative approaches that characterize similarities and differences between the genomes of extant species to reconstruct the genomes of ancestors in a phylogeny that represents the relations between the extant species. On the other hand, the term paleogenomics is also used to

describe the sequencing of ancient DNA from conserved organic material like bones or fossils [96]. In the following section, we aim to give a background on both contexts, before linking both fields to form the central question of this thesis.

1.1.1 Reconstructing ancestral genomes

Genome rearrangements as modifications to the genome sequence have been observed already back in the 1920s. Sturtevant studied the linkage relations in genes of different *Drosophila* species and identified rearrangements by mapping the position of specific genes to their chromosomes [139]. Some years later, Sturtevant and Dobzhansky studied inversions in chromosomes of *Drosophila pseudoobscura* and used the comparison of the gene order between different species to infer the historical relationships between these species [42]. Another 30 years later, in 1963, Pauling and Zuckerkandl introduced the term *paleogenetics* in their studies of human hemoglobin genes [111]. Through manual alignment of specific amino-acid positions in four hemoglobin genes in present-day organisms, they reconstructed likely amino-acids in ancestral polypeptide chains and already discussed perspectives of ancestral reconstruction. In 1965, Camin and Sokal described the principle of evolutionary parsimony, i. e. to explain observed data with as few evolutionary events as possible, to reconstruct phylogenies under the assumption of directed evolution [23], while the principle of minimum evolution has been described by Cavalli-Sforza and Edwards in 1967 [24]. Over the years, additional variants of parsimony emerged as alternative ways to model evolutionary change in the phylogeny (as summarized in [50] and discussed in [51]). However, inferring the phylogeny with the least evolutionary change has been shown to be NP-complete for all basic parsimony variants [38]. On the other hand, as recently summarized in [34], scoring a tree under the different parsimony assumptions is easy and several algorithms have been developed [54, 60, 125].

This brief historical overview illustrates the beginning of ancestral genome reconstruction that is still frequently relying on parsimony as a model of evolution. In the following sections, we give some background on current ancestral reconstruction principles and models for extant and ancient genomes.

Genomes as permutations

Genomes consist of linear or circular nucleotide sequences, called *chromosomes* and *plasmids*. Next to the nuclear genome located in the nucleus of eukaryotes or the nucleoid region in prokaryotes, also organelles such as mitochondria in eukaryotes or plastids

in plants and algae contain DNA as part of the complete genome of an organism. The DNA sequence has two strands, which are oriented in opposite direction to each other. The sequencing of such DNA sequences produces several thousands of reads that are subsequently assembled to retrieve the genome sequence, either fully assembled or in form of several contiguous regions (*contigs*) of the genome.

We abstract from the pure nucleotide sequence by segmenting each chromosome or plasmid into a sequence of non-overlapping, oriented markers. Each marker can then be seen as a substring of the original nucleotide sequence of the genome or the contig, defined by its coordinates and orientation. We assign a unique identifier to each marker from a marker alphabet $\mathcal{M}$. In order to represent the orientation of the marker in the genome (i. e. its location on the homologous strands), we can use a signed representation ($+$ for the forward strand, $-$ for the backward strand) or describe each marker by its extremities. As it is usual in genome rearrangement models, the extremities of a marker are called *head* and *tail*, so a marker a is encoded by the pair (a_t, a_h) or by (a_h, a_t) depending on its orientation.

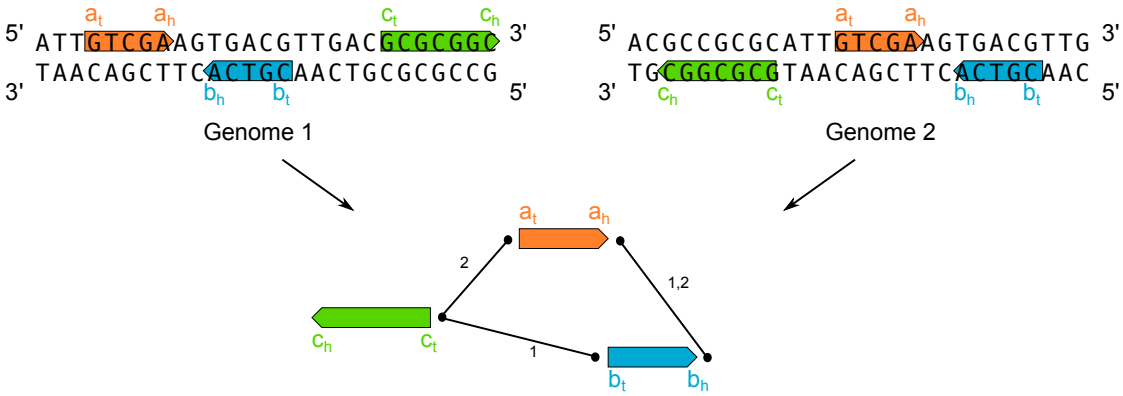


Figure 1.1: Illustration of markers a, b, c defined on substrings of two genomes. Markers are indicated in red, blue and green. Genome 1 is defined by the marker sequence $(..., a_t, a_h, b_h, b_t, c_t, c_h, ...)$, while genome 2 is defined by $(..., c_h, c_t, a_t, a_h, b_h, b_t, ...)$. All adjacencies implied by these marker orders are depicted as a joined graph, where the edges representing adjacencies are marked according to the genome they appear in. The adjacencies $\{c_t, b_t\}$ and $\{c_t, a_t\}$ are conflicting and cannot be part of the same marker order.

Defining such markers on several extant or ancient genomes, we can cluster them into homologous families. Markers belonging to the same family are assumed to be derived from a common ancestor and assigned the same identifier. If each marker family is present in each of the considered genomes, so we do not model the deletion or insertion of markers, we say markers are *universal*. If each marker family is

present only once in each considered genome, so we do not model the duplication of markers, we say markers are *unique*. The loss/gain or duplication of markers influences the complexity of the genome model heavily [53]. In the present work, we assume the basic model of unique and universal markers, hence each considered genome is a permutation of all markers in $\mathcal{M}$. In many publications, markers are coined as "genes", as ortholog gene annotations are a natural source to define marker families [114]. However markers can e. g. also be defined as synteny blocks (as discussed in [124]), extending the analysis of classical gene orders to the genome sequence level. These blocks can be found through multiple alignment of several genomes, which can be computationally expensive for large genomes and hence limiting the number of genomes that can be handled. To avoid this, also more efficient graph approaches have been implemented [95, 145]. If markers should be defined between some fully assembled reference genomes and a set of contigs, a segmentation approach as described in [117, 145] based on the pairwise alignment of contigs onto reference sequences can also be used.

Instead of modeling each genome as a permutation of markers explicitly, we can also define a set of *adjacency* relations on each genome. Assume two markers (a_t, a_h) and (b_t, b_h) to be contiguous in a marker sequence, such that there is no other marker defined between them. Then an adjacency is an unordered pair of the two neighboring marker extremities. Depending on the orientation of both contiguous markers, we can have four different combinations of marker extremities that form the adjacency: $\{a_h, b_t\}$, $\{a_h, b_h\}$,$\{a_t, b_h\}$ and $\{a_t, b_t\}$. A set of adjacencies that can be ordered into a linear or circular marker sequence is said to be *consistent*. However adjacencies are not independent instances: If two adjacencies assigned to the same genome contain the same marker extremity, the set of adjacencies cannot be ordered to a linear or circular marker order. These adjacencies are said to be *conflicting*. Conflicting adjacencies indicate support for different marker orders that need to be evaluated based on the given information, e. g. the phylogenetic context. We assume assembled genomes to be consistent, while reconstructed ancient genomes can contain conflicting adjacencies as an interim result, however most reconstruction methods aim to reconstruct consistent genomes in the end.

The set of adjacencies for one genome naturally defines a graph, where nodes represent marker extremities and edges represent adjacencies. If we also add edges between extremities from the same marker, this graph corresponds to the *breakpoint graph* as described in [115], but we do not explicitly require these edges in our graph representation. Conflicting adjacencies then correspond to branching nodes in the graph. An

example of markers defined on two genomes and a resulting joined graph for both sets of adjacencies is given in Figure 1.1. The graph contains two conflicting adjacencies that cannot be part of the same genome.

Genome distances

Mutations are the driving forces in evolution, applying permanent changes to the nucleotide sequence of a genome. They are rare events scaled to the length of genome sequences and mostly result from damage to the DNA and subsequent erroneous repair mechanisms, or are caused by the dynamic properties of mobile genetic elements. Differences in the genome sequence are the basic level to distinguish organisms and even strains from the same species, so precise models are needed to measure the distance between genomes. Modifications happening to a genome can be classified as either local or global. Local modifications are so called *point mutations*, i. e. nucleotide substitutions, insertions and deletions. They affect single or few base positions in a genome and can be silent due to the degeneracy of the genetic code, hence not influencing the translation of the nucleotide sequence at all. The edit distance between two genome sequences considering these three kinds of local mutations can be efficiently computed with dynamic programming in polynomial time [102, 134].

On the other hand, global modifications affecting larger parts of a genome are called *genome rearrangements*. The set of rearrangement operations includes evolutionary events like sequence inversions, transpositions, translocations, fissions and fusions. They are illustrated in Figure 1.2. Rearrangements are less common than simple nucleotide mutations, hence might be traced back further to the very distant past in the history of related genomes. In addition, they are more closely tied to function, as large-scale rearrangements are much more likely to destroy the contiguity of operational units in the DNA.

Several models have been defined that include some or all rearrangement operations. The now standard Double-Cut-and-Join (DCJ) model [152] subsumes all operations susceptible to alter genomes globally. Efficient algorithms to compute the DCJ distance between two genomes have been developed [11], however including such a complex model into bigger contexts has been proven difficult [140], as genome rearrangements do not comply with the assumption of independence between different parts of the genome. In this thesis, we rely on a simpler model that describes the set of rearrangement operations indirectly: the Single-Cut-or-Join (SCJ) distance introduced by Feijão and Meidanis in 2011 [46]. It is a set-theoretic variant of the breakpoint

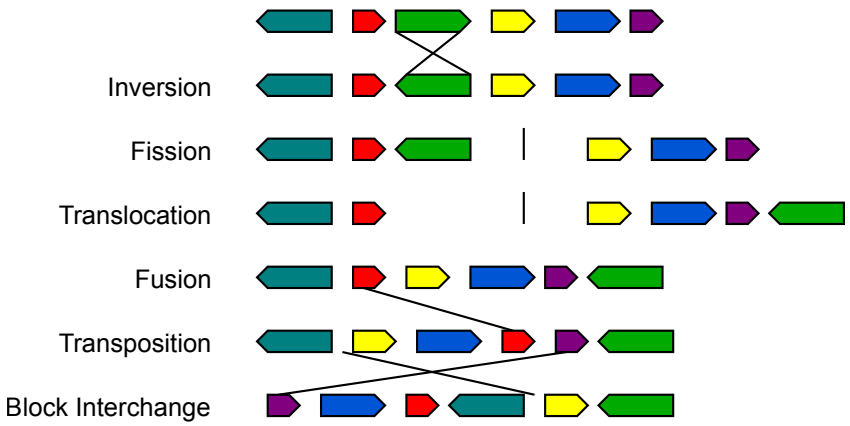


Figure 1.2: Global genome rearrangement operations that reorder blocks of the genome as included in the Double-Cut-and-Join (DCJ) model.

distance [140] for multichromosomal genomes that models *cuts* and *joins* of marker adjacencies.

Definition 1 (Single-Cut-or-Join distance (SCJ))**.** *Given two genomes defined by sets of adjacencies A and B, the SCJ distance between these genomes is*

$$d_{SCJ}(A, B) = |A - B| + |B - A|.$$

In other words, an optimal transformation from genome A to B under the SCJ model can be described by cutting all adjacencies only present in the adjacency set of genome A and joining all adjacencies only present in the set of genome B, resulting in a shortest sequence of cuts and joins defining the rearrangement scenario [46]. While this model is less complex than the DCJ model, which can explicitly describe the course of evolution, it provides the possibility of tractable algorithms in problems where DCJ is too complex, e. g. minimizing the distance between more than two genomes in the context of a phylogeny. We will define this context in the next section.

Phylogeny and Parsimony

Trees are an important data structure in several computer science disciplines. In phylogenetics, trees are used to depict the speciation history of several species during the course of evolution.

Phylogeny Let $G = (V, E)$ be a graph defined by a set of vertices/nodes V and a set of edges E, where each edge connects two vertices in the graph. An edge $e = \{u, v\} \in E$ connecting two vertices u and v is said to be *incident* to both vertices. The degree of a vertex $deg(v)$ is then the number of edges incident to v. An *undirected* graph is a graph in which edges have no orientation, hence each edge is an unordered pair of two vertices in V. If edges are represented by ordered pairs of vertices, the graph is *directed*. For an oriented edge (u, v), node u is the *parent* of v and v is a *child* of u. A *path* in a graph is defined as a set of edges that connects two vertices in V, potentially including several other vertices in between. A path is *simple* if it contains no repeated vertices. In an undirected graph, two vertices $u, v \in V$ are called *connected* if there is a path between u and v. A *connected graph* is a graph in which all vertices are pairwise connected. A *cycle* in a graph is a path where the first and last vertices are the same.

Definition 2 (Tree). *A tree $G = (V, E)$ is a connected graph with no cycles. The set of vertices V is divided into the set of leaves $l \in V$ with $deg(l) = 1$ and the set of internal nodes $i \in V$ with $deg(i) \geq 2$.*

A *rooted tree* contains a designated root node, implying a direction on all edges pointing away from the root. We can then distinguish between the *in-degree* and *out-degree* of a node based on its incident directed edges. The root node has an in-degree of 0. The *depth* of a node v is the length of the simple path from the root to v. If a node u lies on the path from the root to node v, then u is an *ancestral* node of v and v is a *descendant* node of u. A *subtree* rooted at v contains v, all descendants of v, and all edges connecting them. A rooted tree with out-degree of 2 for all internal nodes is called a *binary tree*.

Definition 3 (Phylogenetic species tree). *A phylogenetic species tree $G = (V, E)$ is a tree illustrating the evolutionary relationship of extant and extinct species associated to the leaves $l \in V$. Each internal node $i \in V$ represents a speciation event. If the tree is rooted, the most recent common ancestor to descendant species in the subtree rooted at i is associated to the internal node i.*

In the following, for simplicity we will use the terms *phylogenetic species tree* and *tree* analogously and assume the tree to be rooted. A phylogenetic tree can be *fully-resolved*, in which case the order of all speciation events is known and the tree is binary.

Multifurcating trees on the other hand contain nodes with an out-degree of 3 or more, so called *polytomies*. In addition, phylogenetic trees often have *weighted* edges, where the weight of an edge represents some distance between the species associated to the nodes incident to the edge.

Finding the correct phylogenetic tree for a set of species under consideration is a complex problem that is studied widely. Several methods have been proposed to infer the tree that represents the evolutionary history of these species. Distance-based methods rely on a distance matrix for all considered extant genomes, e. g. based on a pairwise analysis of point mutations between them, whereas likelihood methods build upon probabilistic models to infer the most likely tree topology under all possible tree topologies [52,136]. In this work, we consider the phylogenetic tree to be given, hence concentrating on the following reconstruction problem along the branches of the tree.

Reconstruction Given the speciation history of some species in a phylogenetic tree, we aim to analyze the genomes of ancestral species located at internal points of the tree. Hence considering the diversity of extant genomes and the dynamics of evolution, we can formulate the following central question:

Question 1 (*Ancestral genome reconstruction*)

Given the genome sequences of a set of extant species, how can we infer the genome sequences of ancestral species considering the evolution along the branches of the tree?

Parsimony The assumption of parsimony is the equivalent of Occam's Razor in ancestral reconstruction: given that especially global mutations like genome rearrangements are rare events, it is assumed that the minimal number of changes along the branches of the tree can explain the true evolutionary history of the species involved. We can therefore generally define the problem of ancestral reconstruction of marker orders as an optimization problem minimizing an objective function in the context of parsimony. When we assume the tree to be given, this is known as the *Small Parsimony Problem*.

Definition 4 (Small Parsimony Problem)**.** *Consider a tree $T = (V, E)$ with each leaf l labeled with a set of labels $s_l \subseteq \mathcal{S}$ and a distance function $d : \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{S})$ between sets of*

labels in $\mathcal{S}$. A labeling $\lambda : V \to \mathcal{P}(\mathcal{S})$ over the tree is parsimonious *if $\lambda(l) = s_l \; \forall \, l$ and it minimizes the sum $W(\lambda, T)$ of the distances along the branches of T that is defined as*

$$W(\lambda, T) = \sum_{(u,v)\in E} d\big(\lambda(u), \lambda(v)\big).$$

The set of labels $\mathcal{S}$ can e. g. consist of the set of possible nucleotides or amino acids at specific positions in the genomes, or represent more complex features such as marker orders or sets of adjacencies.

When the tree topology is included in the optimization, the problem is known as the *Maximum Parsimony Problem* (or *Large Parsimony Problem*). Finding the most parsimonious tree is an NP-hard problem for most reasonable distance functions [38].

There exist several variants of the parsimony principle, defining the distance function in Definition 4. We refer to [52] for a detailed review on this topic. Camin and Sokal [23] stated parsimonious evolution as a directed process. They assumed labels to have a logical order, either numerative if labels are numbers or by some qualitative measure otherwise. Then changes of labels along branches of the tree are only allowed in the direction of the given order of labels. To label the tree accordingly, we have to set the label of an internal node u to the minimum label observed at the leaves in the subtree rooted at u.

The framework of *Dollo parsimony* [45, 74] supports the assumption that a label represents a complex characteristic, and hence no label is created twice. In other words, if we characterize each label in a binary $0/1$ format, stating the presence or the absence of a label respectively, then under Dollo parsimony the change $0 \to 1$ along a branch is only allowed once in the tree, while the number of $1 \to 0$ changes is minimized. This excludes the event of *homoplasy*, where two characters evolve independently in two subtrees of the phylogeny. The development of wings in birds and bats is a popular example of homoplasy in nature. So for two labels $x, y \in \mathcal{S}$ and ℓ defined as the number of leaves in the tree, we can define the Dollo parsimony distance as

$$d_{Dollo}(x, y) = \begin{cases} \ell \text{ if } x = 0 \text{ and } y = 1 \\ 1 \text{ if } x = 1 \text{ and } y = 0 \\ 0 \text{ otherwise} \end{cases}$$

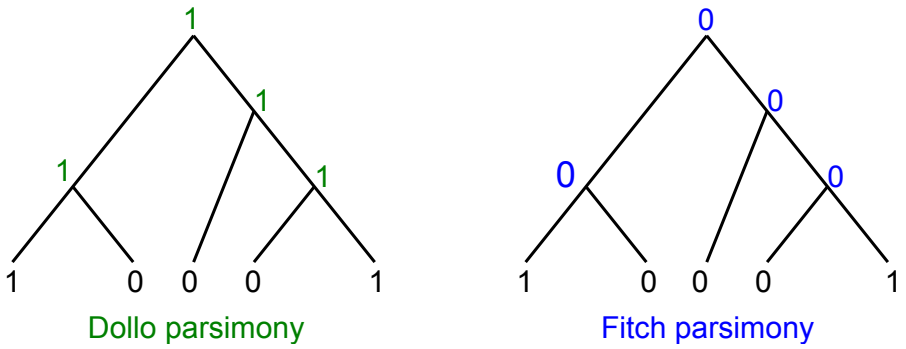


Figure 1.3: Example for optimal labelings under Dollo and Fitch parsimony on a binary label space.

When labels are independent of each other, e. g. , we can assign any two labels $s_1, s_2 \in \mathcal{S}$ to the same ancestor, the parsimonious labeling can easily be found: For any two leaves labeled 1 for a label s, label all vertices on the simple path in the tree connecting these two leaves with 1.

For numerical labels x and y, *Wagner parsimony* defines the distance as the absolute difference $d_{Wagner}(x, y) = |x - y|$. *Fitch parsimony* [54] assumes an equal probability for all changes of labels along the branches of a tree, generalizing Wagner parsimony for binary states to any discrete label alphabet. The Fitch parsimony distance can then be defined as

$$d_{Fitch}(x, y) = \begin{cases} 1 \text{ if } x \neq y \\ 0 \text{ otherwise} \end{cases}$$

In other words, the Fitch model is counting the changes of label states along the branches of the tree, while not assuming an ordering of labels or restrictions to the direction of changes along branches of the tree. An example of tree labelings under Dollo and Fitch parsimony is shown in Figure 1.3. A labeling of a tree under this distance can be computed in polynomial time using a dynamic programming approach. We will revise this in more detail in Chapter 2.

Ancestral reconstruction under the parsimony assumption can be divided into two classes of methods: local and global. For the sake of completeness, we will shortly revise the general strategy of local methods, before we concentrate on the global parsimonious reconstruction based on rearrangement distances. Parts of the following

sections have been included in a recent review on ancestral reconstruction methods [7].

Local reconstruction Local approaches consider the reconstruction of one specific ancestor of interest at a time independently from the other ancestors of the tree [13,29, 84]. Usually, they do not consider an explicit evolutionary model. Given the marker orders for extant genomes, these approaches concentrate on the reconstruction of local syntenic characters as for example adjacencies.

In a first step, these methods compare marker orders of ingroup and outgroup species to define potential ancestral marker adjacencies or intervals. In several methods [68, 106, 117] the set of potential marker adjacencies is based on the Dollo parsimony principle: For a specific adjacency a, if there exist two extant genomes containing this adjacency and their path in the phylogenetic tree contains the ancestor of interest, then a is included in the set of potential adjacencies at this node. Other methods also rely on the Fitch parsimony principle [84]. As adjacencies are not independent, the set of potential adjacencies can subsequently contain conflicting adjacencies due to genome rearrangements, convergent evolution or assembly errors. Hence in a second step, a conflict-free subset of potential ancestral marker adjacencies is selected [88]. The obtained adjacencies are then ordered into so-called Contiguous Ancestral Regions (CARs) [84]. This step is often defined as a combinatorial optimization problem aiming to minimize the number or weight of discarded ancestral adjacencies. It follows principles common in scaffolding methods used to obtain gene orders for extant genomes from sequencing data [20,87].

The software *ANGES* [68] applies these two steps while also considering common intervals of markers in the first step, reporting ancestral genome maps as PQ-trees or PC-trees. The method *FPSAC* adapts these steps to locally scaffold ancient contigs, we will refer to it in more detail later. Because local methods concentrate on only one ancestor in the tree, they do not guarantee to solve the Small Parsimony Problem as defined above. Applying these methods to all internal nodes of the tree separately might miss some relations that are only apparent when all ancestors in the tree are reconstructed simultaneously, resulting in a total tree distance that is not minimal.

Global reconstruction Global approaches on the other hand simultaneously reconstruct ancestral gene orders at all internal nodes of the considered phylogenetic tree, generally based on a parsimony criterion within an evolutionary model. While sim-

ple parsimony distance variants as defined above allow efficient methods to find a parsimonious labeling, genome rearrangement scenarios based on complex rearrangement models can give insights into underlying evolutionary mechanisms and assign explicit evolutionary events like inversions and translocations to the branches of the tree. The *Small Parsimony Problem* has been studied with several underlying genome rearrangement models, such as the breakpoint distance, reversal distance or the DCJ distance [5,21,70,156].

For most rearrangement models that do not include duplications, the distance between two genomes can be computed efficiently. However even the simplest non pairwise ancestral genome reconstruction problem, the median problem [21,123] reconstructing a genome minimizing the distance in a tree with only three leaves, is already NP-hard [140]. Adding duplications makes all problems hard even for the comparison of two genomes [53]. Hence reconstructing rearrangement events that happened along the branches of a tree is not tractable either.

Heuristics for the ancestral genome reconstruction problem usually follow the strategy of assigning an initial genome arrangement to each internal node of the tree and then iteratively refining the solution by solving the median problem for internal nodes until no further improvement in the overall tree distance can be achieved [3]. The algorithm of *GASTS* [151] improves over previous methods applying this strategy by trying to find a good initial arrangement avoiding local optima. Using adequate subgraphs for heuristic assignment of the median, this method can handle multichromosomal data with unique and universal markers.

Another approach is based on the *Pathgroup* data structure [155,156] storing partially completed cycles in a breakpoint graph for each branch in the phylogeny. Informally, the breakpoint graph [9] is a permutation graph showing the relation of a permutation to the identity permutation through differentially colored edges. Graphs are greedily completed and eventually form genomes at all internal nodes. This solution can be used as an initialization prior to local iterative improvements based on the median again using the Pathgroup approach. An interesting property of Pathgroup is that it can handle whole genome duplications.

The method *MGRA* [5] on the other hand relies on a multiple breakpoint graph combining all extant genome organizations into one structure. *MGRA* then searches for breaks in agreement with the species tree structure transforming the breakpoint graph into an identity breakpoint graph. While MGRA originally requires unique and universal markers, it has recently been extended to handle unequal marker content [8]. More complex models of evolution have been considered that include for example

duplications [82, 109], but are tractable only under some specific condition, such as the hypothesis that rearrangement breakpoints are not reused.

The idea underlying the recent tool *RINGO* [47, 48] is to reconstruct intermediate genomes, which are all genomes contained in an optimal pairwise rearrangement scenario between two genomes associated to two nodes in the tree. Intermediate genomes can then be seen as all intermediate steps in the transformation from one genome to another. The method constraints an ancestral genome at an internal node i of the phylogeny to be an intermediate genome of its two child nodes j and k in the tree. This basically cuts the optimal rearrangement scenario between j and k in half: all rearrangements from j to i are assumed to have happened along the branch (i, j) in the tree, the other half is assigned to the other branch, respectively. In addition, the method has been extended to handle unequal marker content heuristically [48].

Some methods adopt a probabilistic point of view, like *Badger* [131], a software using Bayesian analysis under a model where circular genomes can evolve by reversals. It samples phylogenetic trees and rearrangement scenarios from the joint posterior distribution under this model by MCMC implementing different proposal methods in the Metropolis-Hastings algorithm. It is a local search similar to the heuristic on the minimization problem, but instead of giving a single solution without guarantee as an output, it provides a sample of solutions from a mathematically grounded distribution. However it faces the same tractability issues concerning the convergence time.

The method *ROCOCO* [138, 149] is following a more general model by reconstructing ancestral gene clusters based on parsimony and consistency. It follows a variant of the Hartigan algorithm [60] applying a dense or sparse strategy to find a first possibly inconsistent labeling. It then applies efficient methods to identify conflicts in gene clusters and subsequently deleting clusters from an initial labeling to reach consistency.

In comparison to the other tools described above, ROCOCO does not model the evolution of whole genomes as marker permutations. Instead, if the reconstruction is broken down to smaller instances like marker adjacencies – as for the local mapping approaches described above –, tractable exact algorithms can be described.

Definition 5 (Parsimonious Adjacency Labeling Problem)**.** *Let $T = (V, E)$ be a tree with each leaf l labeled with a consistent set of* ***adjacencies*** *$\mathcal{A}_l \subseteq \mathcal{A}$, and d a distance between consistent sets of adjacencies. A labeling $\lambda : V \to \mathcal{P}(\mathcal{A})$ with $\lambda(l) = \mathcal{A}_l$ for each leaf is*

parsimonious *for d if none of the internal nodes $v \in V$ contains a* conflict *and under all consistent labelings λ minimizes the sum $W(\lambda, T)$ of the distances along the branches of T:*

$$W(\lambda, T) = \sum_{(u,v)\in E} d\big(\lambda(u), \lambda(v)\big).$$

The distance d can e. g. be defined as the SCJ rearrangement distance described in Section 1.1.1 modeling the cuts and joins of adjacencies. With this model, the ancestral reconstruction problem becomes tractable. Ancestral genomes that minimize the SCJ distance can be computed efficiently in polynomial time using a variant of the Fitch algorithm [54]. A downside of this approach to reconstruct ancestral genomes is that constraints required to ensure consistency result in mostly fragmented ancestral genomes, i. e. , this reconstruction is sparse and finds only the more fragmented under all co-optimal Fitch solutions. Some adjacencies will be excluded from the reconstructed genomes, although they could be included without causing conflicts neither increasing the SCJ distance along the branches of the tree. As we build upon this basic model in the course of this thesis, we review the Fitch algorithm and the result of [46] in Chapter 2.

In [92], a Gibbs sampler for sampling rearrangement scenarios in a phylogenetic tree under the SCJ model has been described. It starts with an optimal fragmented marker order obtained as described above at each internal node and then explores the space of co-optimal solutions by repeatedly changing the presence/absence scenarios of single adjacencies. However convergence of this sampling method has not been shown.

The *DeCo* [10] framework models the gain and loss of adjacencies in a similar dynamic programming approach. The method assumes the evolutionary history of marker families as additional input, so-called *reconciled gene trees*, depicting speciation and duplication events as well as horizontal and lateral gene transfers and gene losses in the context of the species phylogeny. *DeCo* then aims to reconstruct adjacencies at each ancestral node that are consistent with the respective gene trees for each marker family, however the consistency of the result is not guaranteed. A similar approach has also been explored in the *DUPCAR* algorithm [83]. Several variations of this framework have been developed so far: including the lateral transfers of genes between species [110], handling fragmented extant genome assemblies [6] or adapting a maximum likelihood objective [129]. In addition, *DeClone* [28] models a probabilistic approach by sampling adjacency scenarios according to a Boltzmann probability distribution (see also Chapter 3).

All local and global reconstruction strategies infer tree labelings based on data at the leaves of the tree, taking steps back in time starting from present-day genomes. In the next section, we describe an additional source of information through the study of ancient DNA material.

1.1.2 Sequencing of ancient DNA

Most species that have existed on the planet are extinct, however DNA material can survive up to several hundreds of thousands of years in appropriate conditions [148]. With the invention of the Polymerase Chain Reaction (PCR), it became possible to amplify DNA isolated from archaeological or paleontological remains that can provide direct evidence about the contents and structure of an ancient genome [58]. Early works on ancient DNA (aDNA) concentrated on mitochondrial DNA not older than a few thousand years, recovered for example from quagga [61], extinct moa [32], cave bears [137] or Neandertal [71]. Also museum specimens like the kangaroo rat [141] provided reliable DNA extraction. Later, advances in sequencing technologies and in aDNA recovery protocols [62, 89] opened the way to the sequencing of nuclear aDNA in even older samples of bacteria like *Yersinia pestis* [19, 146, 157] or mammals like the extinct woolly mammoth [93, 116] and ancient horses [103, 104]. We refer to [63] for a review containing many more examples of extinct species that were successfully sequenced by now.

The sequencing of aDNA can shed light on human history besides its genetic evolution. The sequencing data can help to provide evidence of migrations and population development, e. g. human migrations into Europe in the Neolithic [57], or give insights into the molecular mechanisms involved in virulence of human pathogens, e. g. the influenza virus [143] or tuberculosis [17].

However, aDNA research presents extreme technical difficulties for sequencing experiments, because of the small amounts and degraded nature of surviving DNA and the exceptional risk of contamination. When an organism dies, endogenous nucleases normally degrade the sugar-phosphate backbone of its DNA into single molecules. It is often referred to as post-mortem DNA damage [43]. Even if DNA has been conserved in anaerobic conditions, largely neutral pH environments and constant subzero temperatures, usually only short fragments of aDNA can be recovered. Next to the high fragmentation of aDNA samples, nucleotide misincorporation patterns in sequencing have been widely described. The reads from aDNA have been shown to

exhibit specific miscoding lesions, e. g. due to deamination of cytosine residues [107]. In addition, these ancient samples have been exposed to high levels of environmental contamination for a long time, as well as potential contamination when the material is retrieved and processed. Hence there exist many publications stressing the importance of cautious experimental protocols [33, 89] to ensure authenticity and recovery of aDNA, e. g. dedicated, isolated environments to avoid contamination, broad control experiments and reproducability in a second laboratory.

Subsequently, the retrieved sequences are usually aligned to references, and variants are identified keeping aDNA damage patterns in mind [128], precluding the analysis of more complex rearrangements between the ancient and extant genomes [113]. For this, the ancient reads need to be assembled into longer contigs, preferably without a reference sequence guiding the assembly. However, the short length of the aDNA reads usually entails a high number of contigs in the assemblies, even with the help of a reference sequence [19]. We illustrate this problem in our analysis of an aDNA assembly in Chapter 4. So while the contig assembly can be expected to be quite fragmented, classical scaffolding approaches can often not be applied to aDNA data, due to the generally low read coverage and the nature of the aDNA capture process for example. Hence comparative phylogenetic methods following principles similar to the ancestral reconstruction methods described above have to be used to order and orient the obtained contigs.

Question 2 (*Ancient genome scaffolding*)

> *Given a fragmented assembly of ancient DNA reads obtained from conserved genetic material of an extinct species, how can we facilitate a comparison with the genomic sequences of extant related species to reliably scaffold assembled aDNA contigs?*

Yersinia pestis and the bubonic plague

The *Yersinia* genus represents "a key model for understanding the forces that shape the evolution of pathogenic bacteria" [91]. Especially *Yersinia pestis* has been of interest to many researchers as it could be identified as the cause of three major pandemics in the middle ages: the Plague of Justinian, the Black Death in the 14th - 18th century and the Third Pandemic in the 19th century. The bacteria is spreading mostly in rodents and fleas, however in humans it causes the deadly *bubonic plague*. Besides several sequenced and fully assembled extant strains [26], also including the close

relative *Yersinia pseudotuberculosis*, several ancient strains have been sequenced from conserved remains of victims of the bubonic plague [119, 135, 146]. Throughout the following chapters, we concentrate on aDNA data from samples isolated from remains of victims of the Black Death pandemic in the 14th century [19] and the Plague of Marseille in 1720 [18].

The relations in the *Yersinia* phylogeny have been extensively studied [2]. For example, the close relative *Yersinia pseudotuberculosis* is a comparatively non-virulent human pathogen causing a mild disease called *yersiniosis*. The comparative analysis of several strains revealed the pathogenicity of *Yersinia pestis* due to the acquisition of a single protease encoding gene [157]. Further whole-genome comparisons identified a high rate of genome rearrangements induced by a rapid expansion of Insertion Sequence (IS) elements throughout the genome of *Yersinia pestis* [25], while otherwise the genomes in this family are characterized by only a low number of polymorphic nucleotides in comparison to other bacterial pathogens, allowing the confirmation of a unique phylogenetic tree for the *Yersinia* family [1]. This has led to consider the *Yersinia* family as an important model for the study of genome rearrangements during pathogen evolution. Besides the explicit analysis of the genome evolution in *Yersinia*, the genetic information has also been used in a historical context to reconstruct routes of spread of these bacterial pathogens in populations over time, especially shedding light on the source of the three pandemics caused by it [18, 126, 135].

1.1.3 Ancient genome scaffolding and ancestral reconstruction

The problems of reconstructing ancestral genomes on the one hand, and assembling and scaffolding ancient genome sequencing data on the other hand both share similar ground: in a phylogenetic context, we can use the comparison of extant genomes to infer common features that are assumed to be ancestral based on an optimization criterion under an evolutionary model. In this thesis, we want to reconcile both previous questions into a common framework, described as one joint question:

Question 3 (*Integrated phylogenetic assembly*)

Given the phylogeny and genome sequences of a set of extant species, and aDNA sequencing data for one or several extinct species, how can we simultaneously

1. *scaffold the fragmented aDNA assembly through the comparison with extant relatives and*
2. *improve the global reconstruction of the genome sequences of all ancestral species in the phylogeny recognizing the evolution along the branches of the phylogenetic tree?*

The connection between both problems can be seen from two sides. The available aDNA data provides a glimpse into the past and can enable us to improve ancestral reconstructions by strengthening the information that the reconstruction can be built upon besides extant genomes sequenced today. On the other side, as standard scaffolding methods usually cannot be applied to aDNA data due to its fragile nature, a reconstruction based on this sequencing data provides a scaffolding of assembled contigs guided by extant related genomes in the phylogeny.

For extant genomes, there are several methods [6, 66, 69] that use the phylogenetic context for several reference genomes to scaffold a contig assembly. Besides the work presented in this thesis, the only method so far specifically targeted at scaffolding aDNA contigs is FPSAC [117]. It follows a local approach concentrating on one internal node for which aDNA sequencing data is available. It then applies strategies of the described local reconstruction methods by computing copy numbers for markers using discrete parsimony, inferring potential ancestral adjacencies using the Dollo parsimony principle, linearizing these adjacencies and clearing ambiguities due to repeated markers using the algorithms of [118]. The scaffolding itself, aimed at selecting a subset of contig adjacencies compatible with the inferred copy numbers, is achieved through a combinatorial optimization algorithm that does not rely on the given phylogeny [88]. Moreover, as the set of markers is likely not covering the whole ancient genome, gaps between adjacent markers in scaffolds are filled in FPSAC using a multiple alignment of corresponding extant gaps. For each column of the alignment, the parsimonious ancestral state is reconstructed with the Fitch algorithm [54]. Applied to the highly fragmented aDNA contigs of an ancient *Yersinia pestis* strain [19], FPSAC was able to obtain a single scaffold, showing that scaffolding of fragmented ancient genomes can be achieved.

1.2 Thesis overview

The methods and analysis presented in this thesis aim to put this first strategy into a global reconstruction context. Building upon the polynomial time algorithm for the Small Parsimony Problem under the SCJ distance in [46], we first present a global approach for reconstructing all ancestral genomes along a given phylogenetic tree while also scaffolding the aDNA contigs obtained from a preliminary assembly of sequenced aDNA for exactly one internal node of the phylogeny (Chapter 2). While this algorithm still has a polynomial time complexity, we then extend this result to the concept of weighted gene adjacencies being able to include more sequenced aDNA at different nodes in the tree (Chapter 3). The resulting algorithm is an exact, but exponential Fixed-Parameter algorithm, additionally allowing to sample co-optimal solutions.

In Chapter 4, we take the next step, presenting a method to fill the gaps between reconstructed marker adjacencies again drawing on available aDNA data. While this method can be applied to fill the gaps of marker orders reconstructed with the global methods in Chapter 2 and Chapter 3, we also describe a local pipeline building on the gap filling approach to reconstruct marker orders where conflicting adjacencies are solved based on the evidence in the aDNA data directly. This enables us to reconstruct an ancient genome that has the most support by the aDNA reads and to pay special attention to specific genome features hidden in gaps between assembled contigs linked with genome rearrangement breakpoints given as annotations in the extant genomes.

In the last chapter, we evaluate all methods presented in this thesis applied to two ancient *Yersinia pestis* strains in comparison to several extant related species. This allows us to highlight differences and similarities in the proposed reconstructions under the different objectives.

Several parts of this thesis have been published in advance. The theory in Chapter 2 is presented in [77]. Chapter 3 has been published in [80] with an extended version in [79]. The method presented in Chapter 4 together with a part of the analysis in Chapter 5 has been submitted and published as a preprint [78].

Chapter 2

The SCJ Small Parsimony Problem integrating an aDNA assembly

In this chapter, we build upon the result of Feijão and Meidanis in [46] to find a most parsimonious labeling under the SCJ distance. More precisely, we extend the exact small parsimony algorithm described in [46] to the case of multifurcating phylogenetic trees with edge lengths and show how this allows to handle, still within an exact and polynomial time algorithm, constraints from the assembly graph of a sequenced ancestral genome. Part of the theory in this chapter has been published in advance in [77].

Let us briefly recall the considered input. The underlying general data structure is a phylogenetic tree $T = (V, E)$ representing the relations between extant species. The edges of the tree are labeled with lengths describing the evolutionary distances in the tree. We assume that an extant genome at a leaf l is represented by a sequence over a marker alphabet $\mathcal{M}$. Further, this allows us to define a set of adjacencies $\mathcal{A}_l$ for each of these genomes. We denote by $\mathcal{A} = \cup_{l \in V} \mathcal{A}_l$ the union of all different adjacencies observed at any leaf of the tree.

In [46], the classical dynamic programming Fitch algorithm [54] is used over a binary representation of adjacencies to reconstruct ancestral adjacencies that minimize the SCJ distance in the tree. Their result entails three important assumptions:

1. The phylogenetic tree T is binary.
2. The set of adjacencies at each leaf $\mathcal{A}_l$ is consistent.
3. The cost between two labels a and b is $\mathbb{1}_{\{a \neq b\}}$ according to the Fitch parsimony distance.

We will summarize the result in [46] briefly in the next section, before discussing how including ancient DNA data in this polynomial framework can violate assumptions (1) and (2). We then present an extension for the Fitch reconstruction that tolerates these violations at a single node in the tree and hence allows the inclusion of ancient DNA assembly information. The impact of assumption (3) is further studied in Chapter 3.

2.1 Consistent reconstructions using the Fitch algorithm

The Fitch algorithm proceeds in two phases: It first assigns a set B of potential labels to the internal nodes of the tree in a bottom-up traversal, then assigns a final label F to each internal node in a top-down traversal starting at the root node. For a given tree T and a specific adjacency $a \in \mathcal{A}$, the algorithm first labels each leaf l with either $B^a(l) = 0$ or $B^a(l) = 1$ according to the presence or absence of a in the genome associated with l. Then, assuming an internal node u with children v and w, the potential label of u is defined by the potential labels assigned to v and w:

$$B^a(u) = \begin{cases} B^a(v) \cap B^a(w) & \text{if } B^a(v) \cap B^a(w) \neq \varnothing \\ B^a(v) \cup B^a(w) & \text{otherwise} \end{cases}$$

This ensures the principle of parsimony: If a label or in this case an adjacency is seen in both children of a node, it is parsimonious to also consider the presence of this adjacency for the current node. Otherwise, all labels seen in both children have to be considered.

Subsequently, in a top-down traversal of the tree, the algorithm assigns a final label $F^a(root) = b \in B^a(root)$. Notice that if $B^a(root) = \{0, 1\}$, both labels 0 and 1 will result in a labeling of the tree with minimal cost. The final label of any other internal node u of the tree is then unambiguously defined by the final label b of its parent node p:

$$F^a(u) = \begin{cases} b & \text{if } b \in B^a(u) \\ \text{any } b \in B^a(u) & \text{otherwise} \end{cases}$$

In [46], the authors show that with the constraint of choosing 0 at the root in the mentioned case of ambiguity, ancestral genomes at each internal node u defined by the set of adjacencies $\mathcal{A}_u = \{a : F^a(u) = 1\}$ are consistent and minimize the SCJ distance in the tree. The constraint applied ensures tractability and consistency, however it automatically excludes adjacencies that are potentially conflicting, even if they could be included in a consistent genome reconstruction, thus it results in the most fragmented solution at each internal node.

In the following, we first extend the results on the Fitch algorithm to the more general Sankoff-Rousseau algorithm and prove that the optimality and consistency of the reconstruction still hold. Then we show a first approach to include ancient DNA assembly information explicitly in the optimization.

2.2 Generalization to the Sankoff-Rousseau algorithm

Besides reconstructing the most fragmented solution, the Fitch approach as described above is not guaranteed to find all optimal solutions, even if the ambiguity at the root would be explored on the danger of losing consistency. Note that this is not true for the algorithm stated in the original paper by Fitch from 1971. We rely on a generalization of the Fitch algorithm: the Sankoff-Rousseau algorithm [125].

2.2.1 Edge-weighted SCJ Labeling Problem

Like the Fitch algorithm [54], the Sankoff-Rousseau algorithm [125] consists of a bottom-up and a top-down traversal of the tree. However, this more general algorithm induces ambiguity at internal nodes of the tree. For the Small Parsimony Problem with the SCJ distance, it can easily be shown that choosing a 0 label (i. e. the absence of an adjacency) whenever it is possible, also at internal nodes of the tree, results in a consistent labeling, but this could result in an even sparser solution than the result of the Fitch algorithm. Conversely, always including an adjacency in case of ambiguity can result in complex conflicts and would therefore require a subsequent conflict clearing step that is mindful of the tree structure. To avoid this, we propose to include edge lengths in the reconstruction and minimize an edge-weighted SCJ distance. The solution is then likely to be unique in practice, as will be illustrated in the evaluation.

Definition 6 (Edge-weighted SCJ Labeling Problem)**.** *Given a tree $T = (V, E)$ with each leaf labeled with adjacencies and each edge $e \in E$ labeled with an edge length $\ell(e)$, a labeling λ of the internal nodes of T is an* edge-weighted SCJ minimizing consistent labeling *if none of the internal nodes contains a conflict and under all consistent labelings it minimizes the edge-weighted SCJ tree distance*

$$D(\lambda, T) = \sum_{(u,v)\in E} \frac{d_{SCJ}(\lambda(u), \lambda(v))}{\ell\big((u, v)\big)}.$$

2.2.2 Overview of the Sankoff-Rousseau algorithm

The Sankoff-Rousseau dynamic programming algorithm [125] solves the general Small Parsimony Problem for discrete characters. Let S be the set of all possible labels of a node in the phylogeny. For each node v in the tree, the cost of a label $l \in S$ at this node is defined as the minimal total cost within the subtree rooted at v when labeling it with l. It can be computed by minimizing the sum over the cost to all possible labelings of children of v together with the corresponding cost along the edges from v to its children. Then for each node v with children set $d(v)$ in the tree, the cost $C_l(v)$ of assigning label $l \in S$ to this node is defined recursively as follows

$$C_l(v) = \sum_{u\in d(v)} \min_{m\in S}\big(C_m(u) + d(l, m)\big).$$

This equation defines a dynamic programming algorithm whose base case is when v is a leaf in which case $C_l(v) = 0$ if v is labeled with l and $C_l(v) = c_\infty$ for a sufficiently large c_∞ otherwise. Then the cost for each label at each node can be computed in a bottom-up approach, labeling a node as soon as all its children are labeled. Afterwards, we can choose a label with the minimum cost at the root node r as its final assignment $F(r) = \min_l C_l(r)$. In a top-down traversal of the tree, the final label of an internal node v being a child of node w already labeled with $F(w)$ then corresponds to the labels that yielded the minimum in the bottom-up computation, such that

$$F(v) = \min_{l\in S}\big(C_l(v) + d(F(w), l)\big).$$

We refer to [34] for an extensive review on the Sankoff-Rousseau algorithm. The distance d in the algorithm is character dependent, hence we can define a specific distance for each pair of labels along an edge in the tree. If for example labels correspond to nucleotides at a specific position in a genome sequence, we can define a different distance between purines and pyrimidines to emphasize substitutions along an edge within the same base group.

2.2.3 Sankoff-Rousseau on adjacencies with edge lengths

Consider the reconstruction for one adjacency a in a tree T. The set of all possible labels $S = \{0, 1\}$ is then a binary labeling representing the presence or absence of a at an internal node. A leaf is labeled according to the absence or presence of a in the corresponding extant genome. Hence when v is a leaf, we have $C_l^a(v) = 0$ if a is present at v and $C_l^a(v) = c_\infty$ otherwise. The length of the edge between two incident nodes in the tree is then directly included in the bottom-up assignment of the cost $C_l^a(v)$ of assigning label $l \in S$ to v with children set $d(v)$:

$$C_l^a(v) = \sum_{u \in d(v)} \min_{m \in S} \big(C_m^a(u) + d(l, m) \big),$$

$$d(l, m) = \begin{cases} 0 & \text{if } l = m \\ \frac{1}{\ell(v,u)} & \text{otherwise} \end{cases}$$

At the root node r, we choose $F(r) = \min_l C_l^a(r)$ as its final assignment. Then the final label of v being a child of node w is

$$F^a(v) = \min_{l \in S} \big(C_l^a(v) + d(F(w), l) \big).$$

If either at the root node or at an internal node the cost for both $l \in S$ is minimal, we choose the absence of the adjacency to ensure consistency. However with non-trivial edge lengths directly included in the recursion, this ambiguous case with equal cost should rarely occur. Subsequently, the labeling in the top-down phase of the algo-

rithm is already determined by the bottom-up labeling. Hence in most real instances, we can expect that there will be a unique most parsimonious labeling for all adjacencies in practice.

2.2.4 Reconstructing consistent genomes

We show that the edge-weighted Sankoff-Rousseau algorithm assigns consistent genomes. We assume a sparse variant of the algorithm where the label 0 is chosen during the top-down phase any time there is an ambiguity, i.e., either at the root node or at an internal node, the cost for both $l \in S$ is minimal. We call it the *sparse edge-weighted Sankoff-Rousseau algorithm*.

Lemma 1. *Given two conflicting adjacencies a and b, for each node x of T labeled according to the edge-weighted Sankoff-Rousseau algorithm, we have* $C_1^a(x) - C_0^a(x) \geq C_0^b(x) - C_1^b(x)$ *and* $C_1^b(x) - C_0^b(x) \geq C_0^a(x) - C_1^a(x)$ *if there is no leaf l with* $C_1^a(l) = 0 < C_0^a(l) = c_\infty$ *and* $C_1^b(l) = 0 < C_0^b(l) = c_\infty$.

Proof. The proof is by induction on the height h of a node x in the tree, which is the maximal number of nodes on the path from x to any descendant leaf. For $h = 0$, the node is a leaf in the tree consistently labeled as required by the lemma. Table 2.1 indicates all potential labelings for adjacencies a and b and shows that the lemma holds for all leaves.

Table 2.1: Possible leaf assignments for conflicting adjacencies a and b and resulting values for C_1 and C_0. In all cases, the lemma holds.

a \ b	1	0	
1	conflicting	$C_1^a = 0, C_1^b = c_\infty$ $C_0^a = c_\infty, C_0^b = 0$	$C_1^a - C_0^a = C_0^b - C_1^b$
0	$C_1^a = c_\infty, C_1^b = 0$ $C_0^a = 0, C_0^b = c_\infty$	$C_1^a = c_\infty, C_1^b = c_\infty$ $C_0^a = 0, C_0^b = 0$	
	$C_1^a - C_0^a = C_0^b - C_1^b$		$C_1^a - C_0^a > C_0^b - C_1^b$

When $h \geq 1$, we assume that any node with height $g < h$ and therefore all children of x satisfy the lemma. Let $d(x)$ be the set of children of x. For each $y \in d(x)$, we set $k_y := \frac{1}{\ell(x,y)}$.

Assume a leaf labeling for the presence or absence of adjacency a, then the edge-weighted Sankoff-Rousseau algorithm computes the cost of labeling x with 1 or 0 as

$$\begin{aligned} C_1^a(x) &= \sum_{\substack{y \in d(x), \\ C_1^a(y) \geq C_0^a(y)+k_y}} \left(C_0^a(y) + k_y\right) \quad + \sum_{\substack{y \in d(x), \\ C_1^a(y) < C_0^a(y)+k_y}} C_1^a(y) \\ &= \sum_{\substack{y \in d(x), \\ C_1^a(y) \geq C_0^a(y)+k_y}} \left(C_0^a(y) + k_y\right) \quad + \sum_{\substack{y \in d(x), \\ C_1^a(y) < C_0^a(y)+k_y \\ C_0^a(y) < C_1^a(y)+k_y}} C_1^a(y) \quad + \sum_{\substack{y \in d(x), \\ C_0^a(y) \geq C_1^a(y)+k_y}} C_1^a(y) \end{aligned} \tag{2.1}$$

$$\begin{aligned} C_0^a(x) &= \sum_{\substack{y \in d(x), \\ C_0^a(y) \geq C_1^a(y)+k_y}} \left(C_1^a(y) + k_y\right) \quad + \sum_{\substack{y \in d(x), \\ C_0^a(y) \leq C_1^a(y)+k_y}} C_0^a(y) \\ &= \sum_{\substack{y \in d(x), \\ C_0^a(y) \geq C_1^a(y)+k_y}} \left(C_1^a(y) + k_y\right) \quad + \sum_{\substack{y \in d(x), \\ C_0^a(y) < C_1^a(y)+k_y \\ C_1^a(y) < C_0^a(y)+k_y}} C_0^a(y) \quad + \sum_{\substack{y \in d(x), \\ C_1^a(y) \geq C_0^a(y)+k_y}} C_0^a(y) \end{aligned} \tag{2.2}$$

The conditions for the three terms in Equation 2.1 and Equation 2.2 assign each child y unambiguously to only one sum. Further, since all children fulfill the lemma and hence $C_1^a(y) - C_0^a(y) \geq C_0^b(y) - C_1^b(y)$ and $C_1^b(y) - C_0^b(y) \geq C_0^a(y) - C_1^a(y)$, we can derive the following relationships for potential values C_0 and C_1 for the labeling of both adjacencies:

$$\begin{aligned} &\text{For all } y \text{ with } C_1^a(y) + k_y \leq C_0^a(y) \Leftrightarrow C_1^a(y) + k_y - C_0^a(y) \leq 0, \\ &\text{we have } C_0^b(y) + k_y - C_1^b(y) \leq 0 \Leftrightarrow C_0^b(y) + k_y \leq C_1^b(y) \end{aligned} \tag{2.3}$$

$$\begin{aligned} &\text{For all } y \text{ with } C_1^b(y) + k_y \leq C_0^b(y) \Leftrightarrow C_1^b(y) + k_y - C_0^b(y) \leq 0, \\ &\text{we have } C_0^a(y) + k_y - C_1^a(y) \leq 0 \Leftrightarrow C_0^a(y) + k_y \leq C_1^a(y) \end{aligned} \tag{2.4}$$

$$\begin{aligned} &\text{For all } y \text{ with } C_1^a(y) \leq C_0^a(y) + k_y \Leftrightarrow C_1^a(y) - C_0^a(y) \leq k_y, \\ &\text{we have } C_0^b(y) - C_1^b(y) \leq k_y \Leftrightarrow C_0^b(y) \leq C_1^b(y) + k_y \end{aligned} \tag{2.5}$$

$$\begin{aligned}&\text{For all } y \text{ with } C_1^b(y) \leq C_0^b(y) + k_y \Leftrightarrow C_1^b(y) - C_0^b(y) \leq k_y,\\ &\text{we have } C_0^a(y) - C_1^a(y) \leq k_y \Leftrightarrow C_0^a(y) \leq C_1^a(y) + k_y\end{aligned} \tag{2.6}$$

Based on these observations for all children y, we can now show that the lemma holds for node x. First, we plug in Equations 2.1 and 2.2, and directly embed the subtraction inside each term as follows:

$$\begin{aligned}&C_1^a(x) - C_0^a(x)\\ &= \sum_{\substack{y \in d(x),\\ C_0^a(y) \geq C_1^a(y) + k_y}} (-k_y) \quad + \sum_{\substack{y \in d(x),\\ C_0^a(y) < C_1^a(y) + k_y\\ C_1^a(y) < C_0^a(y) + k_y}} (C_1^a(y) - C_0^a(y)) \quad + \sum_{\substack{y \in d(x),\\ C_1^a(y) \geq C_0^a(y) + k_y}} (k_y)\end{aligned}$$

For all children y contained in the first term, we can apply observation 2.3 to derive a condition on adjacency b. Equivalently, we apply observation 2.4 to the last term und observations 2.5 and 2.6 to the middle term:

$$\begin{aligned}&= \sum_{\substack{y \in d(x),\\ C_0^a(y) \geq C_1^a(y) + k_y\\ C_1^b(y) \geq C_0^b(y) + k_y}} (-k_y) \quad + \sum_{\substack{y \in d(x),\\ C_0^a(y) < C_1^a(y) + k_y\\ C_1^a(y) < C_0^a(y) + k_y\\ C_0^b(y) < C_1^b(y) + k_y\\ C_1^b(y) < C_0^b(y) + k_y}} (C_1^a(y) - C_0^a(y)) \quad + \sum_{\substack{y \in d(x),\\ C_0^a(y) + k_y \leq C_1^a(y)\\ C_1^b(y) + k_y \leq C_0^b(y)}} (k_y)\end{aligned}$$

Again, the conditions on adjacency b sort each child y unambigously into only one of the terms. In fact, removing the condition on adjacency a does not move any child y to another term, as all observations are valid on a children considered in the respective term. Further, since $C_1^a(y) - C_0^a(y) \geq C_0^b(y) - C_1^b(y)$ for all children y, we then know that

$$\begin{aligned}&\geq \sum_{\substack{y \in d(x),\\ C_1^b(y) \geq C_0^b(y) + k_y}} (-k_y) \quad + \sum_{\substack{y \in d(x),\\ C_0^b(y) < C_1^b(y) + k_y\\ C_1^b(y) < C_0^b(y) + k_y}} (C_0^b(y) - C_1^b(y)) \quad + \sum_{\substack{y \in d(x),\\ C_1^b(y) + k_y \leq C_0^b(y)}} (k_y)\\ &= C_0^b(x) - C_1^b(x)\end{aligned}$$

The symmetric case for $C_1^b(y) - C_0^b(y) \geq C_0^a(y) - C_1^a(y)$ can be stated equivalently, by simply exchanging variables a and b. This proves that the inequality holds for x and concludes the proof.

□

Corollary 1. *Given two conflicting adjacencies a and b, for each node x of T labeled according to the edge-weighted Sankoff-Rousseau algorithm, if* $C_1^a(x) + k_x < C_0^a(x)$*, then* $C_0^b(x) + k_x < C_1^b(x)$ *if there is no leaf l with* $C_1^a(l) = 0 < C_0^a(l) = c_\infty$ *and* $C_1^b(l) = 0 < C_0^b() = c_\infty$*.*

Proof. The Corollary follows directly from Lemma 1.

□

Lemma 2. *Given two conflicting adjacencies a and b, for each node x of T labeled according to the edge-weighted Sankoff-Rousseau algorithm, if* $F^a(x) = \{1\}$*, then choosing* $F^b(x) = \{0\}$ *is always possible.*

Proof. Suppose there are internal nodes with value 1 assigned to both a and b. Choose such a node with minimal distance to the root and call it v. Let w be the parent of v and $k_v = \frac{1}{\ell(v,w)}$.

Table 2.2 lists all possible combinations of $C_1^a(v), C_0^a(v)$ and $C_1^b(v), C_0^b(v)$ of v. In case of ambiguity, i. e. both C_1 and C_0 are minimal, choosing $F^{a/b}(x) = \{0\}$ is always possible. Cases that cannot occur according to Corollary 1 are marked in the table respectively. For other cases, either $F^a(x) = \{0\}$ or $F^b(x) = \{0\}$ is assigned independent from the parent assignment. In case $(*)$, we only have $F^a(v) = \{1\}$ and $F^b(v) = \{1\}$ if the parent w of v was already labeled $F^a(w) = \{1\}$ and $F^b(w) = \{1\}$. This, however, contradicts the minimality of the depth of v and therefore concludes the proof.

Table 2.2: All combinations of values C_1 and C_0 for both adjacencies a und b computed in bottom-up traversal according to the edge-weighted Sankoff-Rousseau algorithm.

	$C_1^b(v) + k_v < C_0^b$	$C_1^b(v) < C_0^b + k_v$ $C_0^b(v) < C_1^b + k_v$	$C_0^b(v) + k_v < C_1^b$
$C_1^a(v) + k_v < C_0^a$	Cor. 1	Cor. 1	$F^b(v) = \{0\}$
$C_1^a(v) < C_0^a + k_v$, $C_0^a(v) < C_1^a + k_v$	Cor. 1	$(*)$	$F^b(v) = \{0\}$
$C_0^a(v) + k_v < C_1^a$	$F^a(v) = \{0\}$	$F^a(v) = \{0\}$	$F^a(v) = \{0\}$, $F^b(v) = \{0\}$

□

Theorem 1. *For a rooted tree T with leaves annotated with consistent genomes containing the same set of markers, the adjacency sets $\mathcal{A}_v = \{a : F^a(v) = 1\}$ assigned to all internal nodes v with the sparse edge-weighted Sankoff-Rousseau algorithm are consistent genomes and minimize the edge-weighted SCJ distance.*

Proof. The labeling with the edge weighted Sankoff Rousseau algorithm minimizes the edge-weighted distance for each adjacency observed in any leaf of the tree. According to Theorem 6.3 in [46], including the adjacency a in every node v where $F^a(v) = 1$ builds genomes that minimize the SCJ distance over the tree T. Lemma 2 shows that also with the edge weighted Sankoff Rousseau algorithm no conflicting adjacencies will be assigned to a node v. Therefore assigning the set of adjacencies $\mathcal{A}_v$ to any internal node v in T minimizes the total sum of SCJ cost per edge length.

□

2.3 Integrating aDNA sequencing information

We will now extend the framework of the Sankoff-Rousseau algorithm to include aDNA sequencing information by adding an additional leaf with the aDNA information to the tree. For this, we exploit the property of the algorithm to allow for multifurcating trees by extending the given phylogenetic tree with an additional leaf. Further, we show how to still ensure consistency of the result even though the assumption of consistency in the auxiliary leaf genome is violated.

2.3.1 Augmented phylogenetic tree

Now, we assume that one internal node of T is augmented with an assembly graph $A = (V_A, E_A)$ (defined below and illustrated in Fig. 2.1). We will refer to this augmented node as the *assembly graph node* and to the resulting tree as an *augmented phylogenetic tree* in the rest of this chapter.

An assembly graph is a graph whose nodes are contigs, and edges indicate potential sequences that can join contigs. Such a graph can be obtained from the de Bruijn graph or string/overlap graph created by most assemblers, but also from the comparison with extant genomes [117]. The assembly graph is an important source of information for scaffolding purposes, as paths in the graph are possible substrings of the considered genome, while branches indicate uncertainty about the exact genome sequence (see [40] for example). For our purpose, it is important to notice that branching

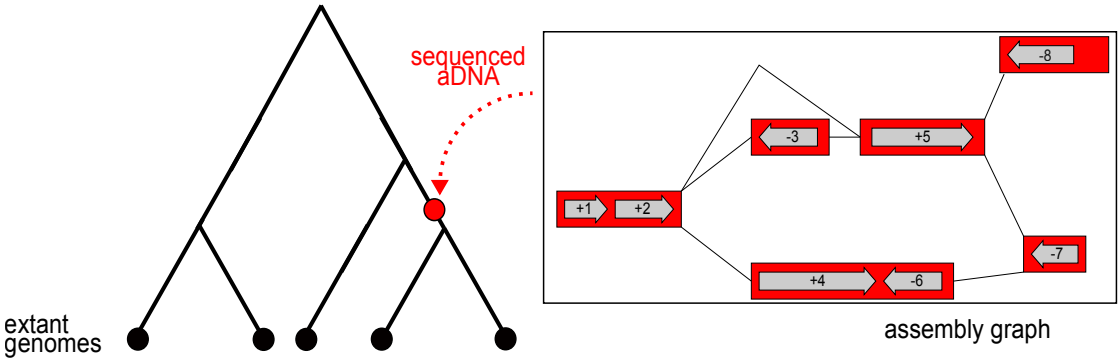


Figure 2.1: Augmented phylogenetic tree annotated with extant genomes at its leaves. One internal node is augmented with an assembly graph illustrating the fragmented assembly. It may contain conflicting adjacencies, e. g. $(2_h, 3_h)$ and $(2_h, 4_t)$, or $(2_h, 3_h)$ and $(2_h, 5_t)$.

nodes in the assembly graph connect one extremity with several others, thus inducing conflicting adjacencies.

Definition 7 (Augmented phylogenetic tree)**.** *Given a tree $T = (V_T, E_T)$ with each leaf labeled with consistent sets of adjacencies. We call the tree augmented if one internal node x is assigned a set of possibly conflicting adjacencies $\mathcal{A}^*$ inferred from an assembly graph G on connected contigs.*

2.3.2 Labeling Problem on an augmented phylogenetic tree

The assembly graph based on ancient sequencing reads defines putative adjacencies between markers on connected contigs (see Figure 2.1). These adjacencies constrain the reconstruction by providing evidence of the genome structure directly seen at an internal point in the tree. This defines the following variant of the Small Parsimony Problem:

Definition 8 (Edge-weighted SCJ Labeling Problem on augmented phylogenetic tree)**.** *Given a phylogenetic tree $T = (V_T, E_T)$ augmented at node x and each edge $e \in E_T$ labeled with an edge length $\ell(e)$, a labeling γ of the internal nodes of T is an* edge-weighted SCJ minimizing consistent labeling that respects the assembly graph *if none of the internal nodes contains a conflict and under all consistent labelings it minimizes the edge-weighted SCJ tree distance and the distance to the adjacencies inferred from the assembly graph:*

$$D(\gamma, T) + d_{SCJ}(\gamma_x, \mathcal{A}^*)$$

In this section, we show how to solve this problem by augmenting the original tree with an additional leaf attached to the assembly graph node. This leaf will be labeled with the presence or absence of an adjacency in the assembly graph just like other leaves representing extant genomes. However the set of adjacencies present in the assembly graph is not necessarily consistent and can cause conflicts. Instead of adding a postprocessing step that resolves all the conflicts in the tree after the reconstruction, in Algorithm 1 we propose an approach that integrates the conflicts resolution into the reconstruction process. To clear conflicts, we rely on the exact polynomial time MAX-ROW-C1P algorithm described in [88]. This algorithm, based on computing a maximum-weight matching in a graph derived from the assembly graph, selects a subset of adjacencies that forms a set of linear and/or circular chromosomes.

Algorithm 1 EWRA: **E**dge-**W**eighted **R**econstruction integrating aDNA **A**ssembly graph

Input: Tree $T = (V, E)$ with edge lengths, extant consistent genomes, aDNA assembly graph at node v
Output: Consistent labeled tree minimizing the edge-weighted SCJ distance

Attach an additional leaf to the assembly graph node v
Assign adjacencies inferred from assembly graph to new leaf node
Reroot the tree such that v becomes its root
for each adjacency a **do**
  for each internal node x in bottom-up traversal of T **do**
    Compute $C_1^a(x)$ and $C_0^a(x)$ with sparse edge-weighted Sankoff-Rousseau
$A = \{a \mid C_1^a(v) < C_0^a(v)\}$
Solve MAX-ROW-C1P for A
for each adjacency a **do**
  for each internal node x in top-down traversal of T **do**
    Compute $F^a(x)$ with sparse edge-weighted Sankoff-Rousseau

Theorem 2. *Given an augmented phylogenetic tree, Algorithm 1 (EWRA) computes an edge-weighted SCJ minimizing consistent labeling that respects the assembly graph in polynomial time.*

Proof. According to Theorem 1, the edge-weighted Sankoff-Rousseau algorithm assigns consistent, SCJ minimizing genomes when the leaf labels are consistent. Rerooting the tree will not affect the outcome of the reconstruction. Then, in the bottom-up

phase, the conflicting leaf will only influence the assignment at the root. All other internal nodes fulfill Corollary 1, as the original leaves are consistently labeled. Therefore they cannot cause a conflicting assignment in the top-down phase when the parent assignment is consistent. As conflicts can be restricted to the root node, they have to be resolved with a minimal increase in parsimony costs before propagating the assignment down the tree during the top-down phase. A maximum cardinality subset of all adjacencies assigned to the root is selected by solving the MAX-ROW-C1P in polynomial time [88]. Note that this set of adjacencies can potentially result in circular scaffolds. With a then consistent root labeling, the top-down assignment will be consistent according to Lemma 2.

□

Hence applying the edge-weighted Sankoff-Rousseau algorithm to the rerooted tree and resolving conflicts at the root introduced by the assembly graph leads to a consistent labeling with minimal parsimony cost and solves the SCJ minimizing consistent labeling problem.

2.4 Evaluation

We first evaluate a reconstruction of a real data set of several *mammalian* genomes with the pure SCJ optimization using the Fitch algorithm [14] compared to a reconstruction with the discussed sparse edge-weighted Sankoff-Rousseau algorithm to measure the differences induced by the change of algorithm and the inclusion of edge lengths in the objective. We further test our method EWRA on a data set of *Yersinia pestis* genomes including ancient DNA sequencing information at one node of the phylogeny in Chapter 5.

Mammalian dataset: Reconstruction in comparison to Fitch

The data set consists of marker orders for several mammalian species as published in [29]. The extant species contain a diverse number of chromosomes ranging from 9 chromosomes in *opossum* to 39 chromosomes in *pig*. Unique and universal markers were computed as synteny blocks from whole-genome alignments with different resolution in terms of minimum marker length. It results in five different data sets varying from 2,185 markers for a resolution of 100 kb to 629 markers for a resolution of 500 kb. We evaluate here the results for the data set with 100 kb, 300 kb and 500 kb resolution.

The underlying phylogeny and applied edge lengths are depicted in Figure 2.2 taken from [29].

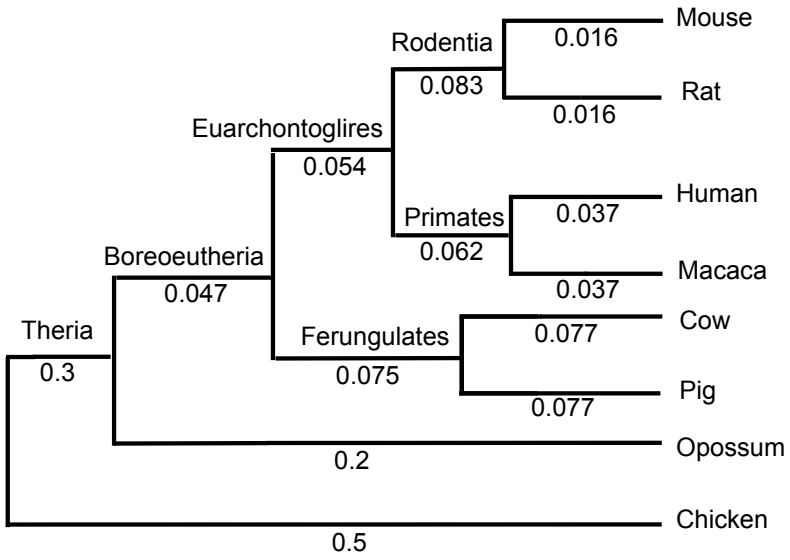


Figure 2.2: Underlying phylogeny for the mammalian data set.

On all three data sets, we computed three reconstructions: (1) using the Sankoff-Rousseau algorithm without considering the lengths of the edges, (2) with the Fitch algorithm implemented in [14] and (3) with our implementation of the edge-weighted Sankoff-Rousseau algorithm as described above. Note that we do not include any aDNA information in the evaluation of this data set (see Chapter 5 for this).

The number of scaffolds is highest with the Sankoff-Rousseau algorithm when edge lengths are not considered, as depicted in Figure 2.3. The number of scaffolds directly correlates with the number of reconstructed adjacencies, indicating as expected that more adjacencies are absent with the Sankoff-Rousseau algorithm, as we have to extend the constraint for the cases of ambiguity also to the internal nodes of the tree. These are exactly the SCJ-optimal solutions that cannot be found by the Fitch algorithm as stated before. In this sense, when the underlying tree is binary, the most fragmented solution found by the Fitch algorithm can differ from the most fragmented solution found by the Sankoff-Rousseau algorithm. Further, Figure 2.3 shows the number of reconstructed scaffolds when the edge lengths are included in the Sankoff-Rousseau algorithm. For all data sets, the number of scaffolds at all internal nodes is reduced, indicating the inclusion of adjacencies that are always excluded with the Fitch algorithm due to the consistency constraint.

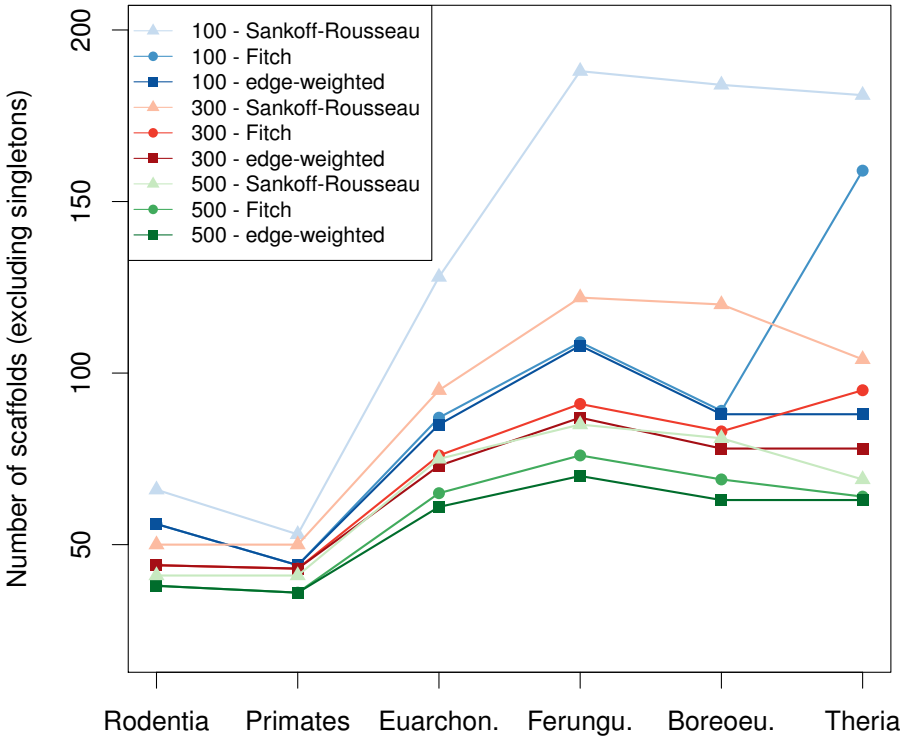


Figure 2.3: Number of scaffolds reconstructed at each internal node for data sets with 100 kb, 300 kb and 500 kb resolution. We ran the Sankoff-Rousseau algorithm without edge lengths (e. g. all edges have a length of 1), the Fitch algorithm as provided in [14] and the edge-weighted Sankoff-Rousseau algorithm.

We directly compared the set of reconstructed adjacencies at each internal node of the tree between the Fitch and the edge-weighted Sankoff-Rousseau algorithm as seen in Figure 2.4. While most adjacencies are reconstructed by both methods, some adjacencies at nodes higher up in the tree are only reconstructed by one of the two methods. To be more precise, between 0.01 and 0.03 percent of all reconstructed adjacencies are only reconstructed by the Fitch approach. This indicates that for such an adjacency, the edge-weighted presence/absence scenario in the tree differs from the pure SCJ scenario, resulting in the exclusion of the adjacency at nodes where it is present in the conservative SCJ scenario. For adjacencies with a mixed signal of presence/absence

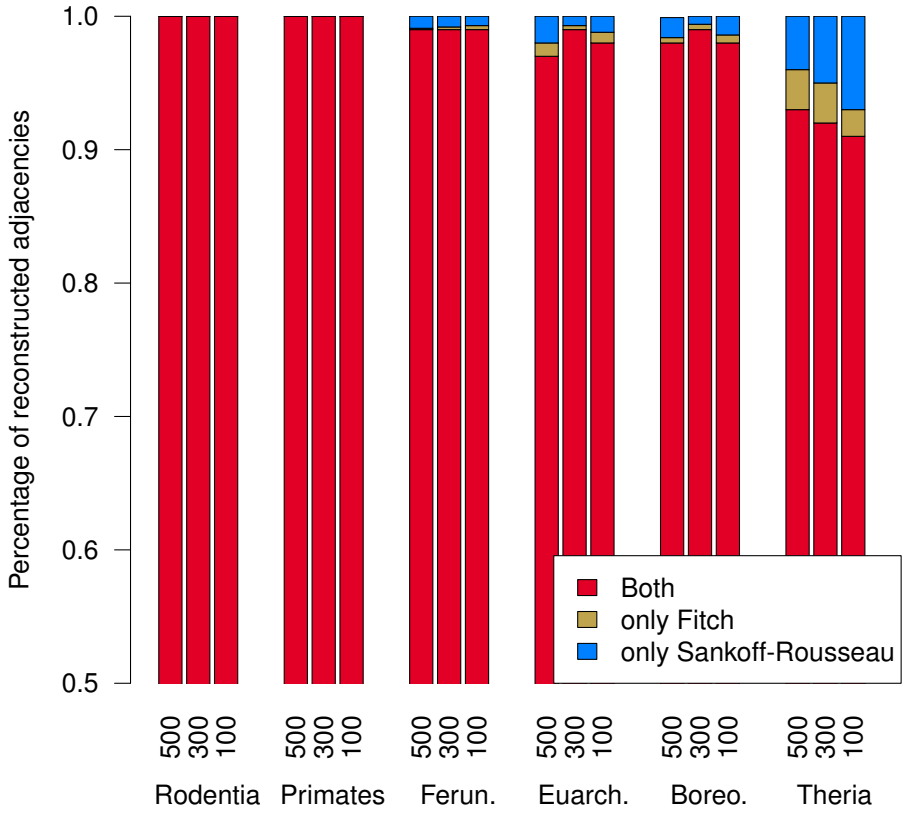


Figure 2.4: Percentage of adjacencies at each internal node for data sets with 100 kb, 300 kb and 500 kb resolution that are reconstructed by both methods, only by the pure Fitch approach and only by the extended Sankoff-Rousseau approach.

in the observed leaf genomes, differences can be caused by the objective function assigning different costs: For two scenarios x and y for the same adjacency, while $d_{SCJ}(x) < d_{SCJ}(y)$, we can have $d_{weightedSCJ}(x) > d_{weightedSCJ}(y)$ while placing changes along edges with greater edge length. Especially in the mammalian phylogeny, as we have two leaves close to the root, the reconstruction for the *Boreoeutherian* ancestor will always be propagated to the *Theria* ancestor and the root node, as the loss or gain of adjacencies will be placed on the long edges to the considered genomes of opossum and chicken, indifferent to the assigned labeling of these leaves. The adjacencies unique to the edge-weighted Sankoff-Rousseau reconstruction are then either part of these

sub-optimal SCJ scenarios, or refer to adjacencies where the edge lengths strengthen the signal for their presence or absence at the root of the tree.

2.5 Discussion

In this chapter, we have described a generalization of the exact algorithm solving the Small Parsimony Problem under the SCJ rearrangement distance. Computing the labeling of internal nodes with the Sankoff-Rousseau algorithm enables the use of multifurcating trees. Including edge lengths still ensures the reconstruction of valid genomes, and it is also expected, in practice, to provide a unique optimal solution under non-trivial edge lengths.

Building upon this result, we presented an integrated phylogenetic assembly approach. It includes aDNA sequencing information in the reconstruction of other ancient genomes in the phylogeny and also scaffolds the fragmented assembly while minimizing the SCJ distance.

Among the questions our work raise, it would be interesting to see if one can extend the current model, that relies on markers that appear once in each genome, in order to integrate copy numbers and unequal marker content. Another question of interest is to design efficient heuristics, or parameterized algorithms, to augment an initial parsimonious consistent labeling with extra adjacencies that preserve both parsimony and consistency.

In our evaluation, we discuss the differences to the pure SCJ reconstruction with the Fitch algorithm, e. g. reducing the fragmentation of reconstructed genomes in the mammalian data set. It also shows the impact of edge lengths in the tree, where the gain or loss of adjacencies is placed at branches to extant leaves with large edge lengths. Especially for adjacencies with a mixed presence/absence signal in the extant genomes, available edge lengths in the phylogeny provide a useful indication to solve ambiguity.

Given the current development in ancient genome sequencing, the limitation of a single ancient genome data set is likely the biggest drawback. Our polynomial time algorithm does not allow for augmenting the phylogenetic tree with more than one assembly graph if a consistent result is required. One would have to add further post-processing steps to the algorithm to ensure consistency, probably losing exactness of the method. Therefore, in the next chapter, we explore a different way of including the aDNA sequencing data, namely in the form of adjacency weights at internal nodes of the tree based on read mappings.

Chapter 3

The SCJ Small Parsimony Problem for weighted adjacencies

In this chapter, we generalize the problem described in the previous chapter to further allow the inclusion of aDNA information at each internal node of the phylogeny. Most parts of this chapter have been published in advance [79, 80]. While in Chapter 2, we have presented a polynomial method that ensures the assignment of a consistent set of adjacencies at each internal node, consistency can no longer be ensured if more than two leaves in the phylogeny are labeled with inconsistent sets of adjacencies. This regards assembly graph leaves representing an ancient DNA data set as well as not fully assembled extant genomes. We present an exponential time algorithm that overcomes this restriction and provides a more general approach to include aDNA data in the reconstruction.

The motivation for this approach is a combination of the two main strategies that existing ancestral genome reconstruction methods concentrate on and that have already been elucidated in Chapter 1: *Local* approaches on the one hand compare marker orders of ingroup and outgroup species to define potential ancestral adjacencies for one specific ancestor in the tree and then select a consistent subset of these adjacencies to obtain a set of CARs [13, 29, 84]. *Global* approaches on the other hand simultaneously reconstruct ancestral marker orders at all internal nodes of the considered phylogeny, generally based on a parsimony criterion as described in the background chapter within an evolutionary model. While this has been studied with several underlying genome rearrangement models, such as the breakpoint distance or the Double-Cut-and-Join (DCJ) distance [5, 70, 156], the problem is NP-hard for most rearrangement distances [140]. An exception is the SCJ distance, for which linear/circular ancestral

marker orders can be found in polynomial time [46]. However constraints required to ensure algorithmic tractability yield fragmented ancestral marker orders.

The work we present is an attempt to reconcile both approaches. The underlying goal of the local approach is to maximize the agreement between the resulting ancestral marker order and the set of potential ancestral adjacencies, independent of the reconstruction at other nodes of the tree. When applying the local strategy to all ancestral nodes independently, potential ancestral adjacencies with a mixed signal of presence/absence in the extant genomes might lead to a set of non-parsimonious ancestral marker orders. On the other hand, the global approach aims at minimizing the evolutionary cost in the phylogeny and can result in more fragmented ancestral marker orders. Therefore in this chapter, we introduce a variant of the Small Parsimony Problem based on an optimality criterion that accounts for both an evolutionary distance and the difference between the initial set of potential ancestral adjacencies and the final consistent subset of adjacencies conserved at each ancestral node. More precisely we consider that each potential ancestral marker adjacency can be provided with a (prior) non-negative weight at every internal node. The contribution of the discarded adjacencies to the objective function is then the sum of their weights. These adjacency weights can e. g. be obtained as probabilities computed by sampling scenarios for each potential adjacency independently [28] or can be based on ancient DNA sequencing data providing direct prior information assigned to certain ancestral nodes. It follows that the phylogenetic framework we present can then also assist in scaffolding fragmented assemblies of aDNA sequencing data [77, 117].

We prove NP-hardness of the problem variant we introduce and describe an exact exponential time algorithm for reconstructing consistent ancestral genomes under this optimality criterion, based on a mixed Dynamic Programming/Integer Linear Programming approach. We show that this Small Parsimony Problem variant is Fixed-Parameter Tractable (FPT), with a parameter linked to the amount of conflict in the data. Moreover, this also allows us to provide an FPT sampling algorithm for co-optimal solutions, a problem recently addressed in [92] using an MCMC approach.

We evaluate our method on a simulated data set and compare our results to several other methods reconstructing ancestral genomes. Further, we apply our method to two real data sets. We analyze the method in terms of complexity of a data set consisting of mammalian genomes spanning roughly one million years of evolution, while analyzing the total tree distance and the fragmentation of the resulting scaffolds. In Chapter 5, we also apply our method to a data set of several *Yersinia pestis* genomes spanning 20,000 years of evolution, allowing us to compare different weighting ap-

proaches for ancestral adjacencies and the impact of including the aDNA sequencing information in the reconstruction. We show that we can reduce the fragmentation of ancestral marker orders in both data sets by integrating adjacency weights while reconstructing robust ancestral genomes.

3.1 Generalization by weighting adjacencies

In this section, we generalize the objective defined in Chapter 2. Again, a phylogeny T is given and we assume that all extant genomes are represented as marker orders or sets of adjacencies respectively. We define the set of all adjacencies observed at some leaf of the tree as $\mathcal{A}$. When considering an internal node v, we define node u as its parent node in T.

Following the spirit of the local reconstruction methods, we first assign a set of potential adjacencies $\mathcal{A}_v \subseteq \mathcal{A}$ to each ancestral node v. It can follow one of the parsimony principles describe before, thus restricting the set of potential adjacencies to a subset of $\mathcal{A}$, or it can just be equal to $\mathcal{A}$ in order to not exclude any adjacency beforehand. None of the potential adjacency sets is required to be consistent. Each $\mathcal{A}_v$ can be seen as a graph assigned to each node v of T, where nodes represent marker extremities and edges indicate potential adjacencies. We will refer to this graph as an *adjacency graph* in the following, despite a slightly different usage of this term in [11].

For each adjacency $a \in \mathcal{A}_v$, we are given a weight $w_{v,a} \in [0,1]$ representing a confidence measure for the presence of adjacency a in species v associated with the edges in its respective adjacency graph. In order to receive a consistent subset of adjacencies in $\mathcal{A}_v$, local methods would choose a maximum-weight subset of adjacencies for a final labeling of node v. In the following, we want to embed this local approach in a global reconstruction, such that in the phylogenetic context the loss of an adjacency with a higher weight along an edge has a higher impact in our objective than the loss of an adjacency of lower weight. We will later illustrate two approaches to obtain these adjacency weights.

Formally, we define two additional variables for each adjacency $a \in \mathcal{A}$ at each internal node $v \in V$: The status of a at node v is represented by $p_{v,a} \in \{0,1\}$, where 0 is associated with the absence and 1 is associated with the presence of the respective adjacency. The variable $c_{v,a} \in \{0,1\}$ indicates a change for the status of an adjacency along an edge (u,v), i.e., $p_{u,a} \neq p_{v,a}$. We consider the problem of optimizing the following objective function, where $\alpha \in [0,1]$ is a convex combination factor.

Definition 9 (Weighted SCJ Labeling Problem)**.** *Let $T = (V, E)$ be a tree where each leaf l is labeled with a consistent set of adjacencies $\mathcal{A}_l \subseteq \mathcal{A}$ and each adjacency $a \in \mathcal{A}$ is assigned a given weight $w_{v,a} \in [0, 1]$ for each node $v \in V$. A labeling λ of the internal nodes of T with $\lambda(l) = \mathcal{A}_l$ for each leaf is an* optimal weighted SCJ labeling *if none of the internal nodes $v \in V$ contains a conflict and under all consistent labelings it minimizes the criterion*

$$D(\lambda, T) = \sum_{v,a} \alpha(1 - p_{v,a})w_{v,a} + (1 - \alpha)c_{v,a}.$$

This objective combines the global optimization according to the SCJ model on adjacencies in the tree with a local weighting that punishes the local loss of adjacencies along an edge. Note that with this criterion, we also sum the weight if an adjacency stays lost over several internal nodes. Depending on the combination factor α, it would hence favor the inclusion of an adjacency over a smaller SCJ cost in the tree if the resulting set of adjacencies is still consistent.

To examine different co-optimal rearrangement scenarios that can explain evolution toward the structure of extant genomes, a sampling method is important. Especially for adjacencies with a mixed signal of presence and absence in the extant genomes, exploring co-optimal solution scenarios can give a more precise notion of how likely the adjacency is present at each internal node of the tree. We thus state the corresponding co-optimal sampling problem.

Definition 10 (Weighted SCJ Sampling Problem)**.** *Given the setting of the Weighted SCJ Labeling Problem, sample uniformly from all labelings λ of the internal nodes of T that are solutions to the Weighted SCJ Labeling Problem.*

We do not take the edge lengths in the tree into account at this point. As we reasoned in the previous chapter, this would likely result in a single optimal solution, whereas here we concentrate more on exploring the whole co-optimal solution space to the problem.

3.1.1 Problem complexity

Aside of the many heuristics for the Small Parsimony Problem for non-SCJ rearrangement models (see for example [70, 151, 156] for the DCJ distance), there exist a few positive results for the Weighted SCJ Labeling Problem for the extreme values of α.

If $\alpha = 0$, we only minimize the gain and loss of adjacencies in the tree and hence the objective function corresponds to the Small Parsimony Problem under the SCJ distance, where a solution can be found in polynomial time [46]. A generalization of this result towards multifurcating, edge-weighted trees including prior information on adjacencies at exactly one internal node of the tree was given in the previous chapter and in [77], respectively.

Recently, Miklós and Smith [92] proposed a Gibbs sampler for sampling optimal labelings under the SCJ model with equal branch lengths. It starts from an optimal labeling obtained as in [46], and then explores the space of co-optimal labelings through repeated constrained parsimonious modifications of the evolutionary scenario for a single adjacency. This method addresses the issue of the high fragmentation of internal node labelings, but convergence is not proven, and so there is no bound on the computation time.

If $\alpha = 1$, i.e., we do not take evolution in terms of SCJ distance along the branches of the tree into account, we can solve the problem by applying independently a maximum-weight matching algorithm at each internal node [88]. This problem is also polynomial for the case of adjacencies.

So the extreme cases of the problem are tractable, and while we assume that the problem is hard for all $0 < \alpha < 1$, it has been proven only for a small range of α.

Theorem 3. *The Weighted SCJ Labeling Problem is NP-hard for any* $1 > \alpha > 33/34$.

The detailed proof for this theorem can be found in [79]. The hardness of the Weighted SCJ Labeling Problem is shown by reduction from the Maximum Intersection Matching Problem, which is defined as follows. Let G_1 and G_2 be two graphs on the same vertex set. Find a perfect matching in G_1 and G_2 such that the number of edges common to both matchings is maximized. NP-hardness of this problem is shown by reduction from 3-Balanced-Max-2-SAT (see [79] for details).

Theorem 4. *The Maximum Intersection Matching Problem is NP-complete.*

The relation of the Weighted SCJ Labeling Problem and the Maximum Intersection Matching Problem can be sketched as follows. For a given instance of the Maximum Intersection Matching Problem, G_1 and G_2, we construct a tree that contains the edges

of both graphs as potential adjacencies. Then for $\alpha > 33/34$, an optimal labeling of two internal nodes corresponds to perfect matchings in G_1 and G_2. Maximizing the number of common edges of the matching minimizes the SCJ distance between the nodes. While this bound on α is quite small, we are confident that the bound can be extended with a more complex reduction from the Maximum Intersection Matching Problem.

3.2 Methods

In order to find a solution to the Weighted SCJ Labeling Problem, we first show that we can decompose the problem into smaller, independent subproblems. Then, for each subproblem containing conflicting adjacencies, we show that, if it contains a moderate level of conflict, it can be solved using the Sankoff-Rousseau algorithm [125] with a complexity parameterized by the size of the subproblem. For a highly conflicting subproblem, we show that it can be solved by an Integer Linear Program (ILP).

3.2.1 Decomposition into independent subproblems

Given the framework of the Sankoff-Rousseau algorithm reviewed in Subsection 2.2.2, we can apply it to solve the Weighted SCJ Labeling Problem. We first introduce a graph that encodes all adjacencies present in at least one internal node of the considered phylogeny. As introduced previously, we consider a tree $T = (V, E)$ where each node is augmented with an adjacency graph.

Definition 11 (Global adjacency graph)**.** *Given an adjacency graph for each internal node v in the tree T. The set of vertices V_{AG} of the global adjacency graph AG consists of all marker extremities present in at least one of the adjacency graphs. There is an edge between two vertices $a, b \in V_{AG}$ that are not extremities of a same marker, if there is an internal node in the tree T whose adjacency graph contains the adjacency $\{a, b\}$. The edge is labeled with a list of all internal nodes that contain this adjacency.*

Each connected component C of the global adjacency graph defines a subproblem composed of the species phylogeny, the set of marker extremities equal to the vertex set of C, and the set of adjacencies equal to the edge set of C. Connected components are generated by rearrangements (the cut or join of an adjacency) and conflicts in the

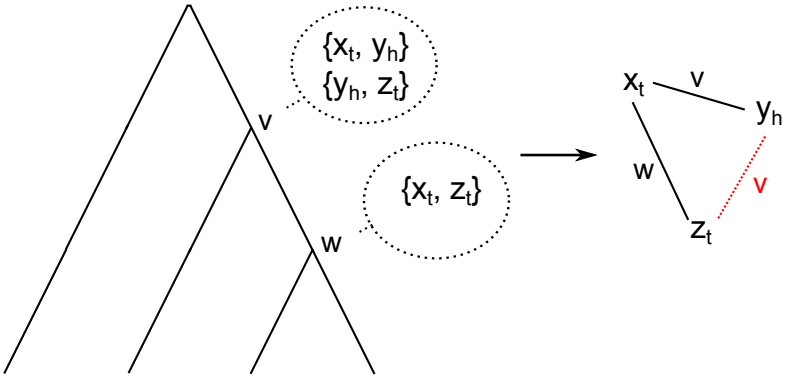


Figure 3.1: Simplified example of a connected component in the global adjacency graph. All internal nodes of the tree are augmented with adjacency graphs. At node v, the graph contains the adjacencies $\{x_t, y_h\}$ and $\{y_h, z_t\}$, at node w, the graph contains the adjacency $\{x_t, z_t\}$. Along the edge $\{w, v\}$, one adjacency has been cut, the other has been joined. In the global adjacency graph, we see a connected component that contains two edges incident to x_t.

set of potential adjacencies. Assume every node x in a connected component being connected to d_x edges. If no two of these edges are labeled with the same species v, then there is no conflict in the connected component. At each internal node v, the marker extremity x is part of only a single adjacency and the connected component represents valid genomes. However if there are two edges incident to x labeled with the same species v, the connected component contains a conflict for v. In this case we need to choose an adjacency for x that is part of the optimal solution for the Weighted SCJ Labeling Problem in respect to the tree. Consistent adjacencies that are never cut or joined will form components with only a single edge. An example for a simple connected component is shown in Figure 3.1.

According to the following Lemma it is sufficient to solve the Weighted SCJ Labeling Problem for each such component independently.

Lemma 3. *The set of all optimal solutions of the Weighted SCJ Labeling Problem is the set-theoretic Cartesian product of the sets of optimal solutions of the instances defined by the connected components of the global adjacency graph.*

Proof. The two extremities of one marker do not have to be part of the same connected component, as the SCJ distance is defined in terms of cuts and joins of adjacencies between marker extremities. As we do not consider the creation of adjacencies that are never observed at any leaf of the tree, all potential adjacencies are contained in

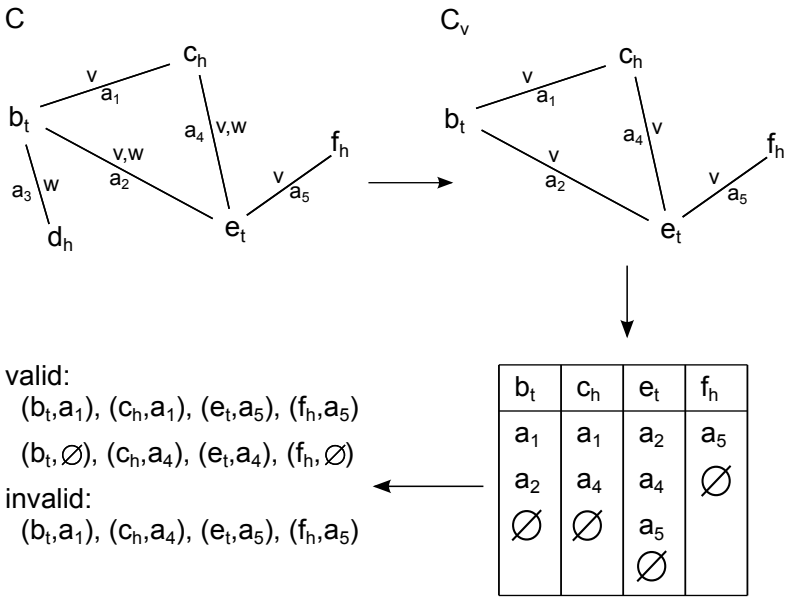


Figure 3.2: Example for a given connected component C. At node v of the tree, possible assignments are defined by the connected component containing all edges annotated with v. Possible assignments for a marker extremity, for example b_t, are defined by the incident adjacency edges, hence in this example we can assign a_1,a_2 or $\emptyset$ to b_t. Here two valid joint labels for v are shown, while the third one assigns different adjacencies for b_t and c_h and is therefore invalid.

the global adjacency graph. We can therefore find a solution of the weighted SCJ distance labeling problem for each connected component of the global adjacency graph independently of each other. □

To solve the problem defined by a connected component C of the global adjacency graph containing conflicts, we can rely on an adaptation of the Sankoff-Rousseau algorithm with exponential time complexity, parameterized by the size and nature of conflicts in C. Hence the algorithm can manage all subproblems that contain only a moderate amount of conflicts, however breaks if there is a single subproblem that is too complex.

3.2.2 Application to the Weighted SCJ Labeling Problem

After labeling each internal node with a set of potential adjacencies (Algorithm 2, line 1), we solve the problem defined by a connected component C of the global adjacency graph (Algorithm 2, lines 2-3).

We define a label of an internal node of the phylogeny as the assignment of at most one adjacency to each marker extremity. More precisely, let x be a marker extremity in C, v an internal node of T, and $e_1, \ldots, e_{d_x}$ be all edges in the global adjacency graph that are incident to x and whose edge label contains v (i.e., represent adjacencies in the adjacency graph of node v). We define the set of possible labels of v as $L_{x,v} = \{\emptyset, e_1, \ldots, e_{d_x}\}$, representing all adjacencies that can be assigned to the marker extremity x. The set of potential labels L_v of node v is then the Cartesian product of the label sets $L_{x,v}$ for all $x \in V(C)$, resulting in a set of discrete labels for v of size $\prod_{x \in V(C)}(1 + d_x)$ (Algorithm 2, lines 5-6).

Note that not all of these joint labelings are valid as they can assign an adjacency $a = \{x, y\}$ to x but not to y, or adjacency $a = \{x, y\}$ to x and $b = \{x, z\}$ to z thus creating a conflict. Figure 3.2 illustrates an example for a simple connected component. For an internal node v, we can reduce the component to all edges labeled with v, as all other edges are not a potential adjacency for this node. We can then simply enumerate all adjacencies in addition to the empty label that each marker extremity can be contained in. Figure 3.2 shows two possible valid labels. The third label is invalid, as the adjacency $a_1 = \{b_t, c_h\}$ is only assigned to one of its contained extremities.

For an edge (u, v) in the tree, we can then define a cost matrix that is indexed by pairs of labels of L_u and L_v, respectively. The cost is infinite if one of the labels is not valid, and defined by the distance in our objective function otherwise. We can then apply the Sankoff-Rousseau approach to find an optimal labeling of all internal nodes of the tree for each connected component C (Algorithm 2, lines 7-11).

Note that, if C is a connected component with no conflict, it is composed of two vertices and a single edge, and can be solved in space $O(n)$ and time $O(n)$, with n being the number of leaves in the tree.

3.2.3 Complexity analysis

The time and space complexity of Algorithm 2 is obviously exponential in the size of C. Indeed, the time and space complexity of the Sankoff-Rousseau algorithm for an instance with a tree having n leaves and r possible labels for each node is $O(nr^2)$ and $O(nr)$ respectively [34]. In our algorithm, assuming n extant species, m_C vertices in the global adjacency graph of C and a maximum degree d_C for vertices (marker extremities) in this graph, $(1 + d_C)^{m_C}$ is an upper bound for the size of the label set L_v for a node v. Moreover, computing the distance between two labels of L_v and L_u, where (u, v) is an edge of T, can trivially be done in time and space $O(m_C)$: If both labels are

Algorithm 2 Adapted Sankoff-Rousseau algorithm to find a solution to the Weighted SCJ Labeling Problem

Input: Tree $T = (V_T, E_T)$ with root r, extant set of adjacencies $\mathcal{A}_l$ for each leaf $l \in V_T$
Output: An optimal weighted SCJ labeling in T

```
1: Assign set of potential adjacencies A_v for each internal node v
2: Construct global adjacency graph AG with A_v ∀ v ∈ V_T
3: for each connected component C = (V_C, E_C) in AG do
4:     for each internal node v ∈ V_T in bottom-up traversal do
5:         Enumerate L_v as cartesian product L_{x,v} ∀ x ∈ V_C
6:         for each l_v ∈ L_v in top-down traversal do
7:             Compute C(l_v, v) with Sankoff-Rousseau
8:     Choose F_C(r) = min_{l_r} C(l_r, r)
9:     for each internal node v ∈ V_T do
10:        Compute F_C(v) with Sankoff-Rousseau
```

valid, it suffices to check how many common adjacencies are present in both labels, while deciding if a label is not valid can be done by a one-pass examination of the label. Combining this with the Sankoff-Rousseau complexity yields a time complexity in order of $O(nm_C(1+d_C)^{2m_C})$ and a space complexity in order of $O(nm_C(1+d_C)^{m_C})$.

Given a general instance, i. e. an instance not limited to a single connected component of the global adjacency graph, we can consider each connected component independently (Lemma 3). For a set of N markers and c connected components in the global adjacency graph defining a conflicting instance, we define D as the maximum degree of a vertex and M as the maximum number of vertices in all such components. Then, the complexity analysis above shows that the problem is Fixed-Parameter Tractable (FPT).

Theorem 5. *The Weighted SCJ Labeling Problem can be solved in worst-case time* $O(nN(1+D)^{2M})$ *and space* $O(nN(1+D)^{M})$.

In practice, the exponential complexity of our algorithm depends on the structure of the conflicting connected components of the global adjacency graph. The dynamic programming algorithm will be effective on instances with either small conflicting connected components or small degrees within such components, and will break down for a single component with a large number of vertices of high degree. For such components, the time complexity is provably high and we propose an ILP to solve them.

3.2.4 An Integer Linear Program for complex components

If a connected component is too complex for the DP algorithm, we can formulate the optimization problem as an ILP. We consider two variables for any adjacency a and node v, $p_{v,a} \in \{0,1\}$ and $c_{v,a} \in \{0,1\}$, defined as in Section 3.1.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{a,v} \alpha(1 - p_{a,v}) w_{a,v} + (1-\alpha) c_{a,v} \\
\text{subject to} \quad & p_{v,a} + p_{u,a} - p_{w,a} \geq 0 \text{ for } (w,u),\ (w,v) \in E(T) && (c_1) \\
& p_{v,a} + p_{u,a} - p_{w,a} \leq 1 \text{ for } (w,u),\ (w,v) \in E(T) && (c_2) \\
& p_{v,a} + p_{u,a} + c_{v,a} \leq 2 \text{ for } (u,v) \in E(T) && (c_3) \\
& p_{v,a} + p_{u,a} - c_{v,a} \geq 0 \text{ for } (u,v) \in E(T) && (c_4) \\
& p_{v,a} - p_{u,a} + c_{v,a} \geq 0 \text{ for } (u,v) \in E(T) && (c_5) \\
& -p_{v,a} + p_{u,a} + c_{v,a} \geq 0 \text{ for } (u,v) \in E(T) && (c_6) \\
& \sum_{a=(x_t,y)} p_{v,a} \leq 1 \text{ and } \sum_{a=(x_h,y)} p_{v,a} \leq 1 \\
& \text{for any marker } x \text{ and node } v && (c_7)
\end{aligned}
$$

The first two constraints express c_1 and c_2 basic parsimony assumptions, i. e. , no adjacency can be present at a node w if it is not present in any child node of w. Also, if an adjacency is present in all child nodes of w, it has also to be present in w. Consistency of the solution is ensured with constraint c_7. Constraints $c_3 - c_6$ define the correct value for $c_{v,a}$ dependent on the value of p_a along an edge (u,v), distinguishing between all status combinations for an adjacency along the edge. Minimizing $c_{v,a}$ then ensures parsimony of the solution. This ILP has a size that is polynomial in the size of the problem.

3.2.5 Sampling co-optimal labelings

The Sankoff-Rousseau DP algorithm can easily be modified to sample uniformly from the space of all co-optimal solutions to the Weighted SCJ labeling Problem in a forward-backward fashion. In the bottom-up traversal, in addition to the minimal cost induced by labeling a node v with a specific label $a \in L$, we can also store the number of optimal solutions under this label for the subtree rooted at v. Let x and y be the children of v, and L_x and L_y the sets of labels that induced the minimum value for a

at v (which means a label out of these sets is assigned in the backtracking phase if v is labeled with a). Then

$$\overline{C}(v,a) = \left(\sum_{l \in L_x} \overline{C}(x,l)\right)\left(\sum_{l \in L_y} \overline{C}(y,l)\right)$$

gives the number of optimal solutions for the subproblem rooted at v. At the root, we might have the choice between different labels with minimum cost. Let L_{root} be the set of these labels, then the number of overall possible co-optimal solutions is simply the sum of the number of solutions for all optimal root labels:

$$\overline{C}(root) = \sum_{l \in L_{root}} \overline{C}(root,l).$$

Subsequently in the top-down traversal, choose a label $l \in L_{root}$ with probability

$$\frac{\overline{C}(root,l)}{\overline{C}(root)}.$$

If at an internal node more than one label in a child node induced the minimum value, choose one of these labels analogously. This classical dynamic programming approach leads to the following result with the complexity analysis analogously to Theorem 5.

Theorem 6. *The Weighted SCJ Sampling Problem can be solved in worst-case time* $O(nN(1+D)^{2M})$ *and space* $O(nN(1+D)^{M})$.

For subproblems that are too large for being handled by the Sankoff-Rousseau algorithm, the SCJ Small Parsimony Gibbs sampler recently introduced [92] can easily be modified to incorporate prior weights, although there is currently no proven property regarding its convergence.

3.2.6 Weighting ancestral adjacencies

A first approach to assign weights to ancestral adjacencies follows the idea presented in [28] to consider evolutionary scenarios for an adjacency independently of the other adjacencies in a probabilistic framework. The Boltzmann distribution describes a probability of a system to be in a certain state depending on the energy of that state and

the temperature of the system. In our context, we want to describe the probability of a labeling σ of presence and absence for an adjacency in the tree depending on the parsimony score $p(\sigma)$ of that scenario. The parsimony score is simply the number of gains and losses for the adjacency along the branches of the tree based on σ.

As described in [28], the Boltzmann score for a scenario is then defined as

$$B(\sigma) = e^{-\frac{p(\sigma)}{kT}},$$

where kT is a given constant as the product of the Boltzmann constant and the thermodynamic temperature. For each adjacency $a \in \mathcal{A}$, denote by $\mathcal{S}(a)$ the set of all possible evolutionary scenarios for the adjacency a (i. e. not restricted to parsimonious scenarios). The partition function of a as defined in the Boltzmann context is then given by

$$Z(x, y) = \sum_{\sigma \in \mathcal{S}(a)} B(\sigma),$$

so adding up the Boltzmann scores over all scenarios in $\mathcal{S}(a)$ as the normalizing constant for the probability distribution. Subsequently, the Boltzmann probability for the scenario σ is defined as

$$Pr(\sigma) = \frac{B(\sigma)}{Z(a)}.$$

We can infer the weight of the adjacency at internal node v as the ratio of the sum of the Boltzmann probabilities of all scenarios where the adjacency is present at node v. All such quantities can be computed in polynomial time [28].

The parameter kT can then be used to skew the Boltzmann probability distribution. If kT tends to zero, parsimonious scenarios are heavily favored and the Boltzmann probability distribution tends to the uniform distribution over optimal scenarios, while when kT tends to ∞, the Boltzmann distribution tends toward the uniform distribution over the whole solution space.

In our experiments, we use the tool DeClone [28] to infer adjacency weights by sampling scenarios under the Boltzmann probability distribution. We will refer to these weights as *Boltzmann weights* in the following. We chose a value of $kT = 0.1$ that favors parsimonious scenarios but considers also slightly suboptimal scenarios and $kT = 1$ that samples more evenly over the whole solution space.

When aDNA sequence data is available for one or several ancestral genomes, markers identified in extant species can be related to assembled contigs of the ancestral

genome, as in [117] for example. For an ancestral adjacency, it is then possible to associate a sequence-based weight to the adjacency – either through mapping based methods such as the probabilistic model of GAML [22], or scaffolding methods such as BESST [120] for example. In comparison to the weighting approach described above, these weights are then not directly based on the underlying phylogeny, but provide an external signal for the confidence of adjacencies at the respective internal node. We will refer to them as *aDNA weights* in the following.

The probabilistic model of GAML [22] is based on the mapping of reads to a gap between potentially adjacent markers. The general idea is that an adjacency whose gap can be covered continuously by aDNA reads has a higher probability to be ancestral than an adjacency that is not supported by aDNA reads. To apply this model, we first need to compute potential DNA sequences filling the gaps between two adjacent marker extremities (template gap sequences). For example in FPSAC [117], these templates are obtained by aligning the gap sequences of the corresponding conserved extant adjacencies and reconstructing a consensus ancestral sequence using the Fitch algorithm. While this is a simple and efficient way to provide gap sequences, it mostly works for well conserved extant species.

Then we compute the weights as a likelihood of this putative gap sequence given the aDNA reads. Each adjacency together with its template gap sequence details a proposition for an assembly A as a piece of the real ancestral sequence. Given the aDNA read set R, the model defines a probability

$$Pr(R|A) = \prod_{r \in R} Pr(r|A)$$

for observing the reads R given that A is the correct assembly. The probability $Pr(r|A)$ can be computed by aligning r to the assembly A while the alignment is evaluated under an appropriate sequencing error model. Taking into account L as the length of A, the model defines the probability of an alignment with m matches and s mismatches as $R(s,m)/2L$, where $R(s,m) = \epsilon^s(1-\epsilon)^m$ includes the respective sequencing error rate ϵ. By dividing the matching probability with $2L$, we consider that reads can be aligned to both strands of the assembly sequence. With a set of mappings S_r of a read r to the assembly A, we approximate the probability by

$$Pr(r|A) \approx \frac{\sum_{j \in S_r} R(s_j, m_j)}{2L}.$$

In [22], this model is used to iteratively find a high likelihood assembly or improve existing assemblies.

3.2.7 An extinct leaf

As an extension to the framework described above, we can also assume an ancient genome to be a leaf in the phylogeny, i. e. the ancient species has no direct sequenced descendants. The dynamic programming approach we propose can deal with this naturally. Each leaf that corresponds to an extant genome is assigned a fixed label for each connected component based on the observed adjacencies in the genomes. In contrast, a leaf representing an ancient genome can be treated similar to an internal node: given a set of potential adjacencies for this node, we can enumerate all possible labels for each connected component. These labels then influence the bottom-up labeling in the tree, while an optimal label can be chosen in the top-down refinement. Note however that there are no descendants that could influence the labeling for such a leaf in the tree. For example, in the evaluation in Chapter 5, we include an ancient *Yersinia pestis* strain that has no known extant descendants. We assume the same set of potential adjacencies as for all internal nodes in the phylogeny and weight them with the GAML probabilities based on the available aDNA data. In this approach, we follow the strategy of the local reconstruction methods: we first assign potential adjacencies and then select a consistent subset that minimizes the distance in the tree.

Another approach can be to adapt the strategy of Chapter 2 for extinct leaves with ancient DNA: we constrain the set of potential adjacencies at the beginning to all adjacencies seen in an assembly graph of the ancient data, then use the comparison in the tree to scaffold the assembly by including new adjacencies. However, as the initial adjacency assignment of the extinct leaf influences the reconstruction of the whole tree, it contains the danger of too much missing information: in extant leaves, a missing adjacency is really missing, whereas in ancient leaves, a missing adjacency can also be due to not assembled regions in the genome. We avoid this in Chapter 2 by rerooting the tree.

3.2.8 Implementation

A Python implementation of the adapted Sankoff-Rousseau algorithm is available at `https://github.com/nluhmann/PhySca`. The tool is divided into two steps: First, preprocessing scripts are provided that parse extant marker order files in GRIMM format and extract all potential ancestral adjacencies. Adjacency weights can then

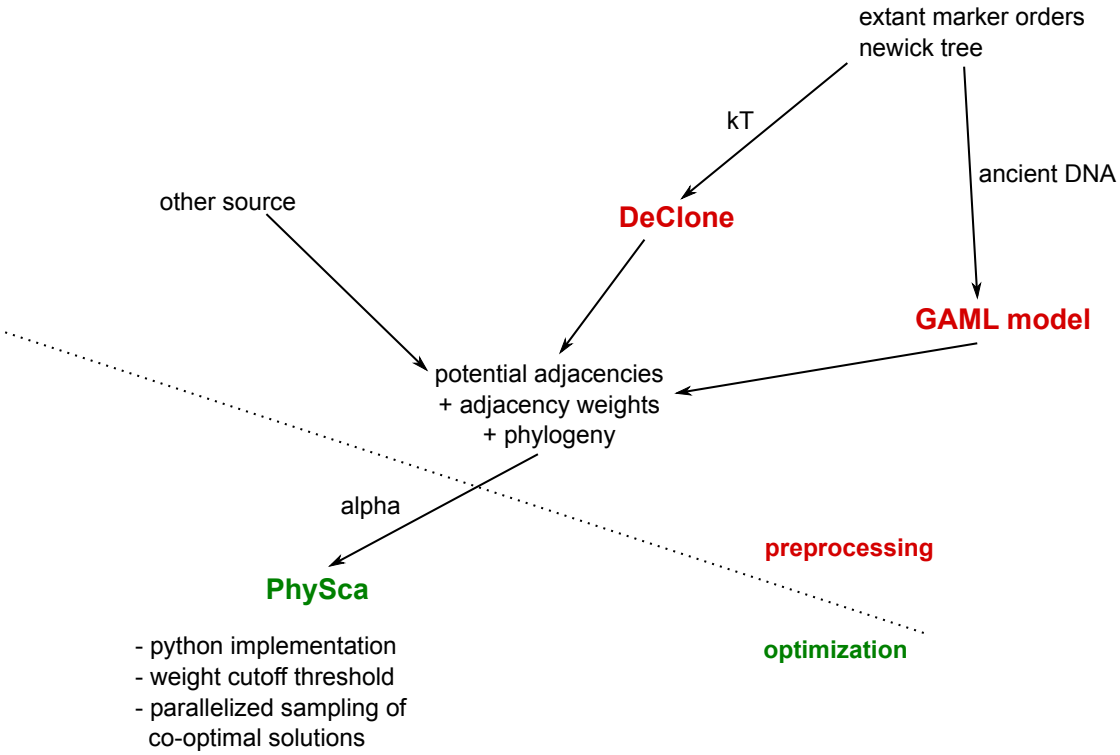


Figure 3.3: The described method is implemented in python and divided into a preprocessing and an optimization phase. The scripts for the preprocessing compute potential ancestral adjacencies for each node together with an adjacency weight. We provide two methods to compute the weights: either sampling adjacency scenarios with DeClone under parameter kT or computing the probability for an adjacency based on ancient DNA read mapping. Further, the user can also provide adjacencies and weights obtained from other sources. The main tool PhySca then takes the thus preprocessed data as input and reconstructs ancestral genomes for a specific value of α. It provides a cutoff parameter x for data sets that are too complex to handle. When x is provided, all adjacencies with a weight smaller than x are not considered as potential adjacencies at an ancestral node. The sampling of co-optimal solutions can be parallelized: either in python itself or with the help of a framing bash script.

be computed using DeClone [28] based on the Boltzmann sampling described above, or based on aDNA read mapping to all potentially ancestral gaps under the GAML model. The main implementation *PhySca* then takes all potential adjacencies with an associated adjacency weight as input, allowing an easy extension to other adjacency weighting approaches. An overview of the implementation and program parameters is given in Figure 3.3.

3.3 Results

We evaluated our algorithm on a simulated data set and compared its sensitivity and precision to several other reconstruction methods. Further, we applied our method to two real data sets: mammalian and *Yersinia* genomes. The results for the *Yersinia* data set are given in Chapter 5. The mammalian data set was used in the studies [29] and [92] and has also been used in the evaluation for Chapter 2. It contains six mammalian species and two outgroups, spanning over 100 million years of evolution, and five different marker sets of varying resolution (minimal marker length). Our experimental results consider issues related to the complexity of our algorithm, the use of a pure SCJ reconstruction (obtained when the α parameter equals 0) and the relative impact of the value of α on both the total evolutionary cost and the ancestral marker orders fragmentation. We refer to Figure 2.2 on page 34 for the species phylogeny of this data set.

3.3.1 Simulations

We created simulated data sets as described in [47]: with a birth-rate of 0.001 and a death rate of 0, we simulated 20 binary trees with 6 leaves and scaled the branch lengths such that the tree has a diameter of $2n$, where n is the number of markers in each unichromosomal genome. The root genome with $n = 500$ markers was evolved along the branches of the tree by applying inversions and translocations with a probability of 0.9 and 0.1 respectively. The number of rearrangements at each branch corresponds to the simulated branch length, the total number of rearrangements ranges from 1242 to 2296 in the simulated trees. We compare results of our implementation *PhySca* for different values of $\alpha \in \{0, 0.3, 0.5, 0.8, 1\}$ with the tools *RINGO* [47], *MGRA* [8], Fitch-SCJ [14], *ROCOCO* [138,149] (dense approach for signed adjacencies) and *ANGES* [68] (adjacencies only). We computed adjacency weights as described in Subsection 3.2.6 with the software *DeClone* [28] and parameter $kT \in \{0.1, 1\}$.

The methods *RINGO* and *MGRA* are global approaches minimizing the DCJ-distance in the tree, while *ANGES* reconstructs specific ancestors locally in the tree and is applied for each node separately. For $\alpha = 0$, our objective is finding a consistent, most parsimonious solution and equals the objectives of Fitch-SCJ and *ROCOCO*, where Fitch-SCJ always finds the most fragmented solution whereas *ROCOCO* and our method aim at reporting least fragmented reconstructions.

By comparing the simulated and reconstructed adjacencies, we can count for each method the number of true positives (TP) as all simulated and correctly reconstructed

adjacencies, false positives (FP) as all falsely reconstructed adjacencies, and false negatives (FN) as all simulated adjacencies that were not reconstructed. We computed the following measures for the performance of the different methods:

$$\text{Sensitivity } S = \frac{TP}{(TP + FN)}$$

$$\text{Precision } P = \frac{TP}{(TP + FP)}$$

$$\text{F-score } F_\beta = \frac{(1 + \beta^2)\, TP}{(1 + \beta^2)\, TP + \beta^2\, FN + FP}$$

A high sensitivity indicates the ability to recover the true marker order of ancestors in the phylogeny, while a high precision denotes few wrongly reconstructed adjacencies. Our method reaches a high precision of 0.99 for all values of $\alpha \geq 0.5$, while increasing the sensitivity in comparison to the pure Fitch-SCJ solution by reducing the fragmentation of the reconstructed scaffolds, as shown in Figure 3.4. For higher values of α, the influence of the weighting becomes apparent: for $kT = 0.1$, the precision only decreases for $\alpha = 1$, while for $kT = 1$, the precision decreases also for lower values of α, however leading to more complete reconstructions. In comparison, both DCJ-based methods *RINGO* and *MGRA* produce less fragmented solutions by recovering more true adjacencies under the jeopardy of also reconstructing more false adjacencies. The sensitivity and precision of Fitch-SCJ, *ROCOCO* and *ANGES* are comparable to our method for low to medium values of α.

The F_1 score (F-score with $\beta = 1$) assesses the relation of sensitivity and precision with equal importance. *RINGO* achieves a better F_1 score than all other methods. The $F_{0.5}$ score (F-score with $\beta = 0.5$) emphasizes the precision of a method over its sensitivity. With this measure, our method with $kT = 1$ and $\alpha = 0.5$ outperforms the other tools, while *ROCOCO* and *ANGES* also reach similarly good scores.

In general, it can be seen that the equal contribution of global evolution and local adjacency weights in the objective function provides a reliable reconstruction. The simulations also underline that our tool is useful tool to explore the solution space under different values of α, and we will investigate this further on a real data set in the next section.

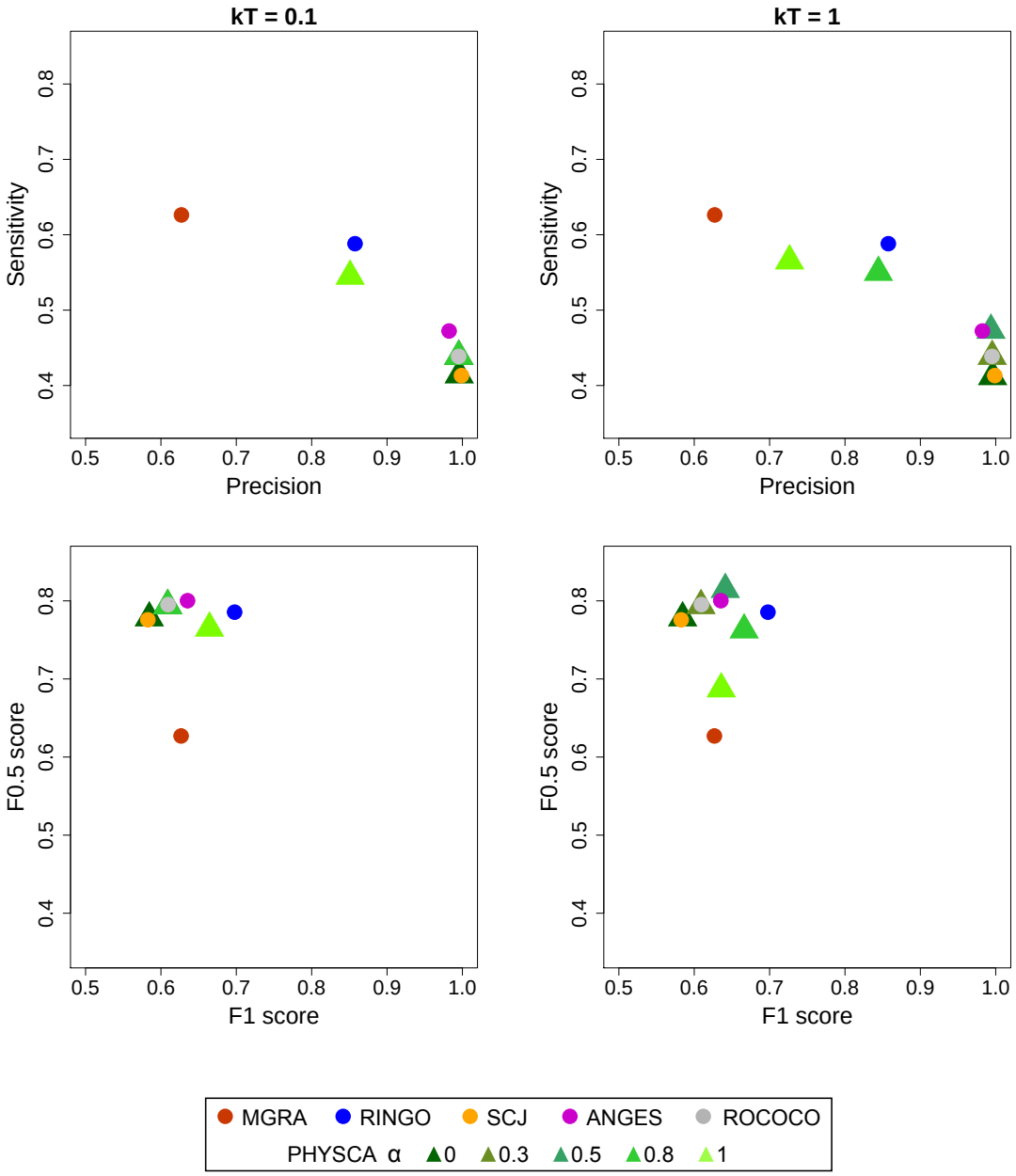


Figure 3.4: Average precision and sensitivity (top), and F_1 and $F_{0.5}$ (bottom) of reconstructions on 20 simulated data sets. Adjacency weights have been obtained with parameters $kT = 0.1$ (left) and $kT = 1$ (right).

3.3.2 Mammalian data set

The mammalian data set has already been used in Section 2.4. It consists of five different data sets varying from 2,185 markers for a resolution of 100 kb to 629 markers for a resolution of 500 kb.

We considered all adjacencies present in at least one extant genome as potentially ancestral. To weight an adjacency at all internal nodes of the tree, we relied on evolutionary scenarios for each single adjacency, in terms of gain/loss, independently of the other adjacencies (i. e. without considering consistency of ancestral marker orders). We considered two values of the DeClone parameter kT, 0.1 and 1, the former ensuring that only adjacencies appearing in at least one optimal adjacency scenario have a significant Boltzmann weight, while the latter samples adjacencies outside of optimal scenarios. For the analysis of the ancestral marker orders obtained with our algorithm, we considered the data set at 500 kb resolution and sampled 500 ancestral marker orders for all ancestral species under different values of α.

Complexity

The complexity of our algorithm depends on the size of the largest connected component of the global adjacency graph. In order to restrict the complexity, we kept only adjacencies whose weights are above a given threshold x. As expected, Figure 3.5 shows the decrease in computational complexity correlated to threshold x for the five different minimal marker lengths. In most cases, all connected components are small enough to be handled by our exact algorithm in reasonable time except for very large components in the marker sets with higher resolution under a low threshold x. For the 500 kb data set with $x = 0.2$ and $kT = 1$, the computation of one solution takes on average 200 s on a 2.6 GHz i5 with 8 GB of RAM. It can be reduced to 30 s when Boltzmann weights are based on $kT = 0.1$. This illustrates that our algorithm, despite an exponential worst-case time complexity, can process realistic data sets in practice.

Optimal SCJ labelings

Next, we analyzed the 500 optimal SCJ labelings obtained for $\alpha = 0$, i. e. aiming only at minimizing the SCJ distance, and considered the fragmentation of the ancestral marker orders (number of CARs) and the total evolutionary distance. Note that, unlike the Fitch algorithm used in [46], our algorithm does not favor fragmented assemblies by design but rather considers all optimal labelings. Sampling of co-optimal solutions

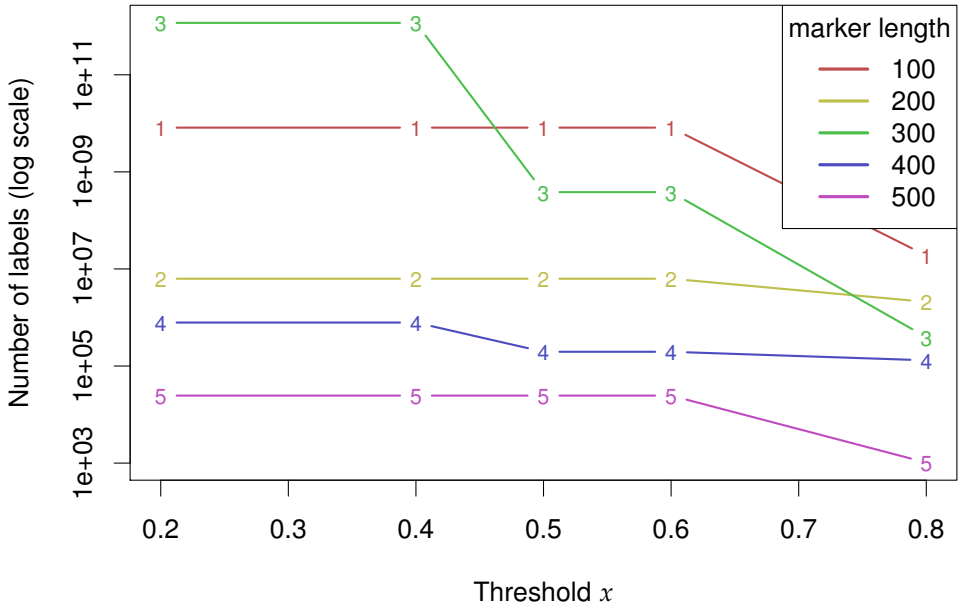


Figure 3.5: Number of different labels for the largest connected component in each of the mammalian data sets. This statistic provides an upper bound for the actual complexity of our reconstruction algorithm.

shows that the pure SCJ criterion leads to some significant variation in terms of number of CARs (Figure 3.6).

In contrast, Table 3.1 shows that most observed ancestral adjacencies are present in all sampled scenarios. Only about 5% of adjacencies, mostly located at nodes higher up in the phylogeny, are present only in a fraction of all sampled scenarios, indicating that there is a small number of conflicts between potential adjacencies that can be solved ambiguously at the same parsimony cost. The optimal SCJ distance in the tree for $\alpha = 0$ is $1,674$, while the related DCJ distance in the sampled reconstructions varies between 873 and 904 (see also Figure 3.8). In comparison, we obtained a DCJ distance of 829 with GASTS [151], a small parsimony solver directly aiming at minimizing the DCJ distance. More precisely, over all ancestral nodes, 70 adjacencies found by GASTS do not appear in any of the leaves and do thus not belong to our predefined set of potential ancestral adjacencies, and another 147 appear in the 500 samples with

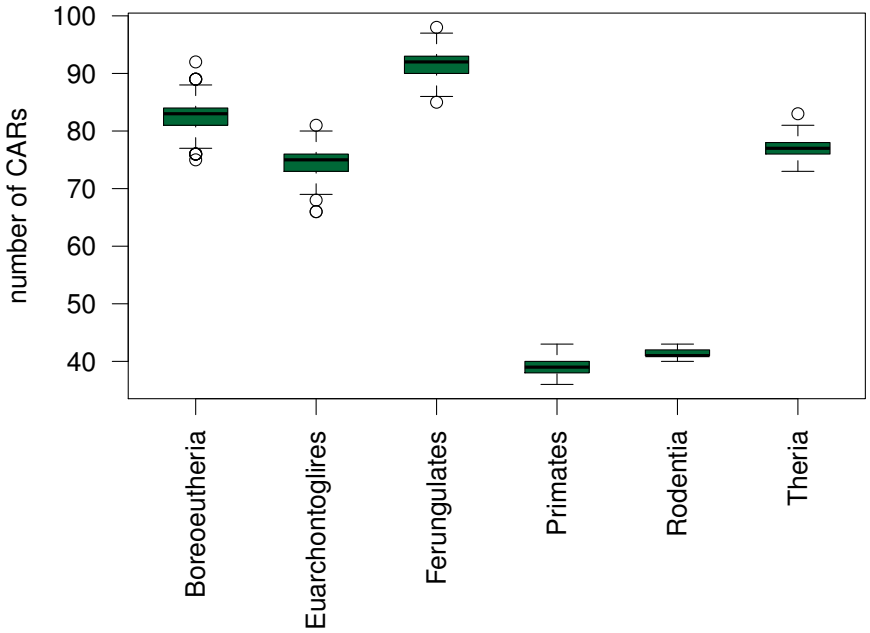


Figure 3.6: Number of reconstructed CARs at each internal node in 500 samples for the mammalian data set with 500 kb resolution, $x = 0.2$ and $\alpha = 0$.

Table 3.1: Frequency of adjacencies in 500 samples with $\alpha = 0$ as percentage of optimal labelings they appear in.

Ancestor	Frequency f		
	$f = 100\%$	$100\% > f > 50\%$	$f < 50\%$
Boreoeutheria	94.66	1.07	4.27
Euarchontoglires	95.42	0.88	3.79
Ferungulates	96.53	0.55	2.92
Primates	98.82	0.34	0.84
Rodentia	99.49	0.34	0.17
Theria	97.67	0.89	1.43
root node	92.23	1.23	6.53

a frequency below 50%. This illustrates both a lack of robustness of the pure SCJ optimal labelings, and some significant difference between the SCJ and DCJ distances.

Finally, we compared the Boltzmann probabilities of ancestral adjacencies with the frequency observed in the 500 samples. There is a very strong agreement for Boltz-

mann weights obtained with $kT = 0.1$ as only 14 ancestral adjacencies have a Boltzmann weight that differs by more than 10% from the observed frequency in the samples. This shows that, despite the fact that the Boltzmann approach disregards the notion of conflict, it provides a good approximation of the optimal solutions of the SCJ Small Parsimony Problem.

Ancestral reconstruction with Boltzmann weights and varying values of α.

For $\alpha > 0$, our method minimizes a combination of the SCJ distance with the Boltzmann weights of the adjacencies discarded to ensure valid ancestral marker orders. Again, we sampled 500 solutions each for different values of α with the 500 kb data set. We distinguish between DeClone parameters $kT = 0.1$ and $kT = 1$. Figure 3.7 shows the impact of this parameter on the distribution of the weights. While for $kT = 0.1$, favoring parsimonious scenarios, we get specific weights grouping the adjacencies according to their extant appearances, with $kT = 1$, sampling more uniformly, we get a much more distributed picture.

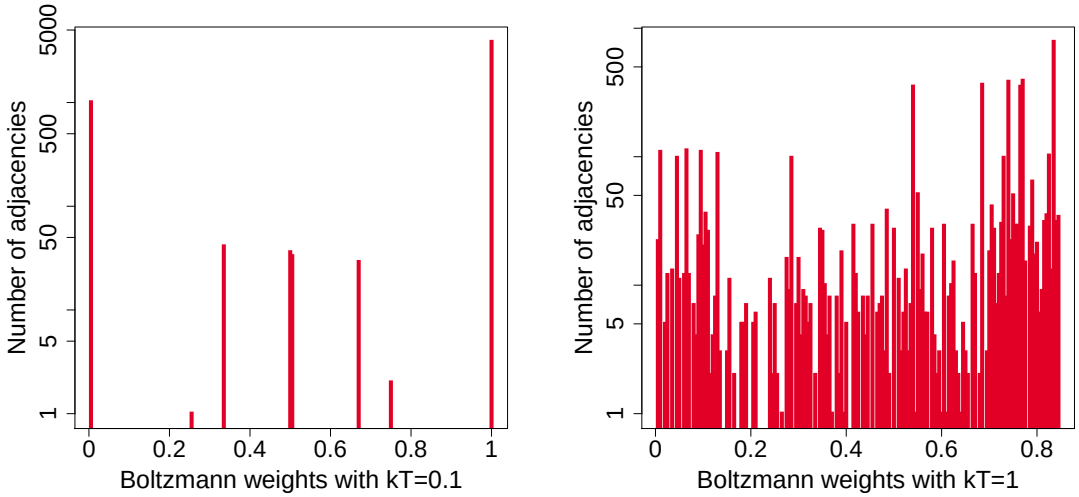


Figure 3.7: Distribution of Boltzmann weights of all potential adjacencies for the 500 kb data set with DeClone parameter $kT = 0.1$ and $kT = 1$.

Figures 3.8 and 3.9 show the respective observed results with the reconstruction based on Boltzmann weights in terms of evolutionary distance and fragmentation.

For $kT = 0.1$, the optimal SCJ and DCJ distance over the whole tree hardly depends on α. Including the Boltzmann weights in the objective actually results in the same

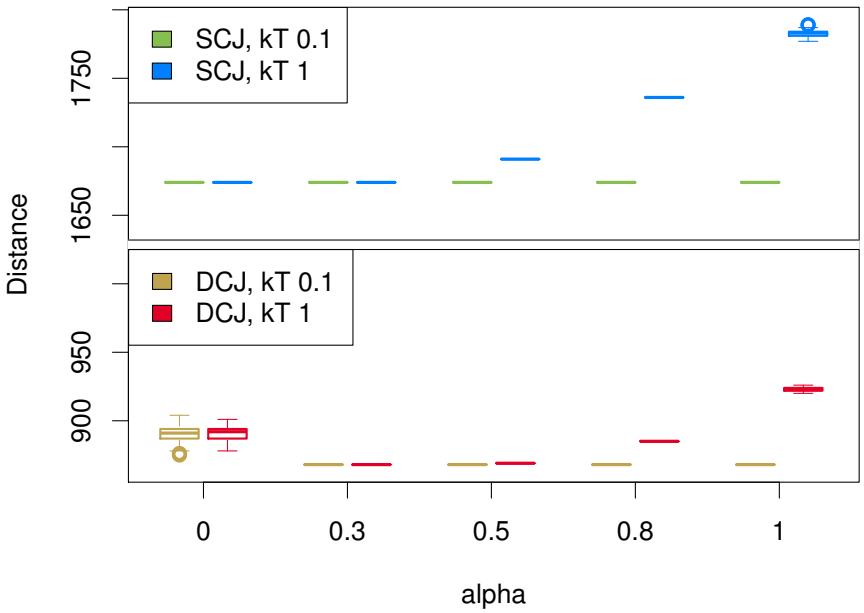


Figure 3.8: SCJ distance (upper half) and DCJ distance (lower half) in the whole tree for all samples and selected values of α in the mammalian data set.

solution, independent of $\alpha > 0$. In fact, while applying a low weight threshold of $x = 0.2$, the set of potential adjacencies is already consistent at all internal nodes except for a few conflicts at the root that are solved unambiguously for all values of α. This indicates that building Boltzmann weights on the basis of mostly optimal adjacency scenarios (low kT) results in a weighting scheme that agrees with the evolution along the tree for this data set. More importantly, Figures 3.8 and 3.9 show that the combination of Boltzmann weights followed by our algorithm, leads to a robust set of ancestral marker orders.

In comparison, for $kT = 1$, we see an increase in SCJ and DCJ distance for higher α, while the number of CARs at internal nodes decreases, together with a loss of the robustness of the sampled optimal results when α gets close to 1. It can be explained by the observation that the weight distribution of ancestral adjacencies obtained with DeClone and $kT = 1$ is more balanced than with $kT = 0.1$ as it considers suboptimal scenarios of adjacencies with a higher probability. It further illustrates that, when the global evolutionary cost of a solution has less weight in the objective function, the

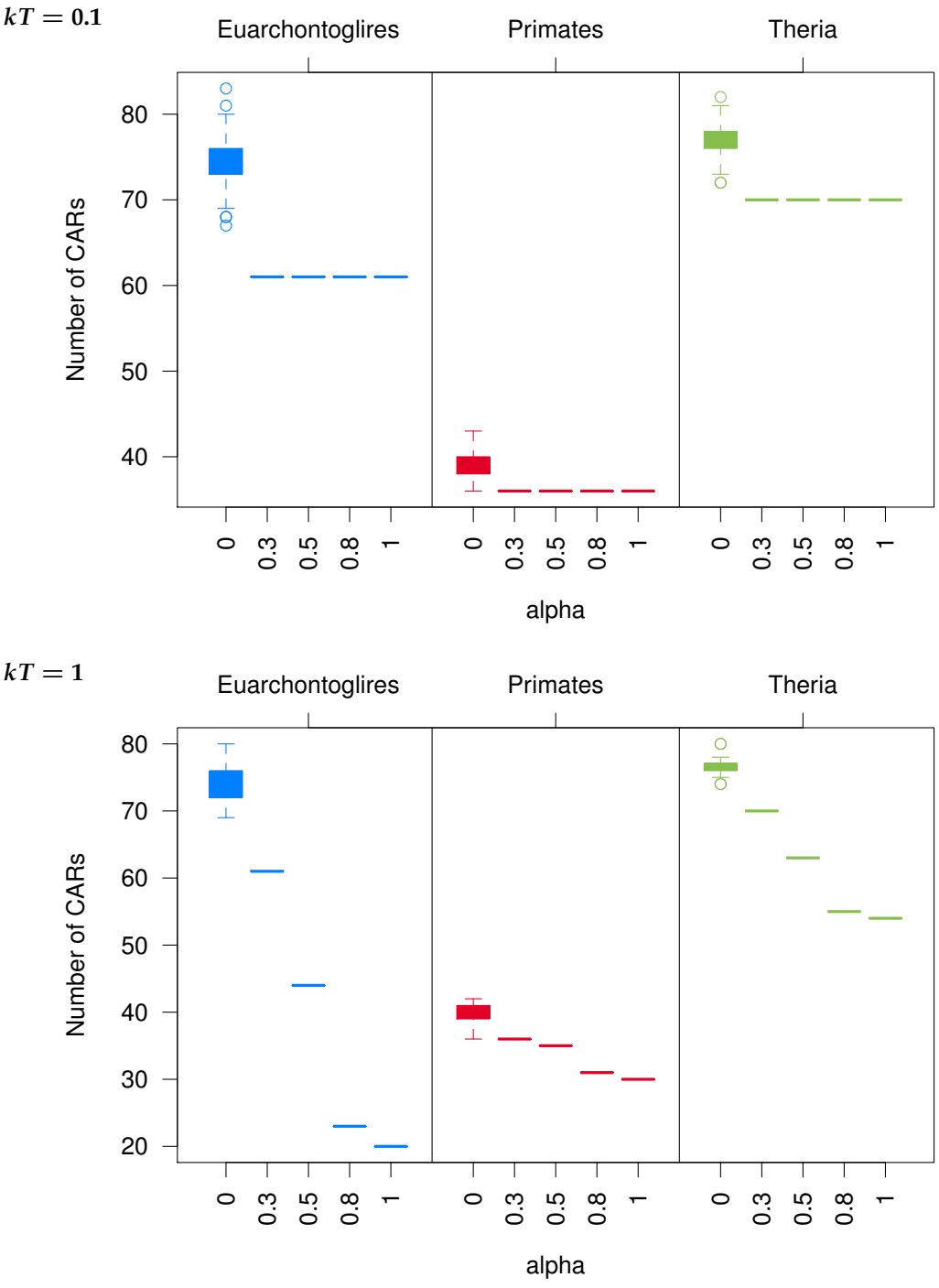


Figure 3.9: Number of CARs in the mammalian data set in all samples at selected internal nodes for different values of α reconstructed with Boltzmann weights under $kT = 0.1$ and $kT = 1$. While the number of CARs differs in the case of $\alpha = 0$ where the adjacency weights are not considered, the fragmentation stays constant for the other values of α.

algorithm favors the inclusion of an adjacency of moderate weight that joins two CARs while implying a moderate number of evolutionary events (for example an adjacency shared by only a subset of extant genomes). From that point of view, our algorithm – being efficient enough to be run on several values of α – provides a useful tool to evaluate the relation between global evolution and prior confidence for adjacencies whose pattern of presence/absence in extant genomes is mixed.

Finally, we can briefly relate the PhySca reconstructions with Boltzmann weights in this chapter to our results with the edge-weighted Sankoff-Rousseau algorithm in Chapter 2. The comparison is summarized in Figure 3.10, depicting the number of adjacencies reconstructed by only some of the methods. We also include the Fitch algorithm as a reference in the comparison, while it basically corresponds to one sampled solution with $\alpha = 0$. As we do not include aDNA data with this data set, the algorithmical difference between PhySca and the edge-weighted Sankoff-Rousseau al-

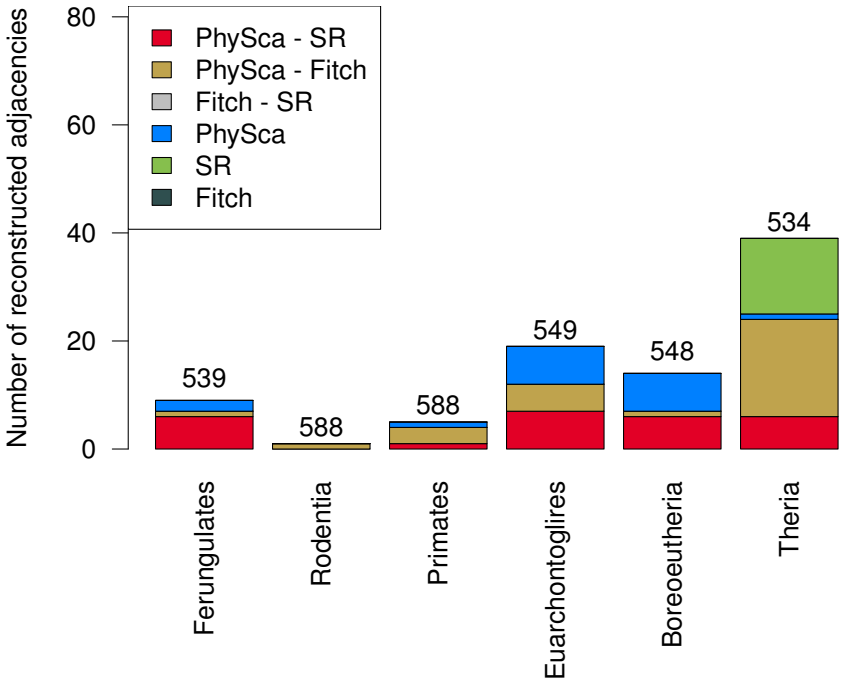


Figure 3.10: For the 500 kb data set, we compare the reconstructions with the pure Fitch algorithm, the edge-weighted Sankoff-Rousseau (SR) algorithm described in Chapter 2 and PhySca with $\alpha = 0.5$. This plot depicts the number of adjacencies reconstructed only by some of the methods, and we indicate the large number of adjacencies reconstructed by all methods at each internal node above each bar in the plot.

gorithm are the included edge lengths on the one hand, while we apply the weight threshold x to reduce the complexity of the data for PhySca on the other hand.

As expected, all of the adjacencies reconstructed by the Fitch algorithm are reconstructed by PhySca. However, we see some adjacencies reconstructed only by PhySca and Sankoff-Rousseau, as well as some adjacencies reconstructed only by PhySca and Fitch. In the first case, these are adjacencies included in the reconstruction reducing the fragmentation, hence they are not reconstructed in the most fragmented Fitch solution. In the latter case, these are differences already observed in Chapter 2 due to the influence of the edge lengths in the reconstruction, being the counterpart of all the adjacencies reconstructed only with the edge-weighted Sankoff-Rousseau algorithm.

Interestingly, we also observe a small number of adjacencies only reconstructed with PhySca. All of these adjacencies have a high adjacency weight, causing the inclusion of the adjacencies at some internal node at the price of slightly increased global tree cost. On the other hand, none of the adjacencies filtered by the threshold x in the PhySca reconstructions are reconstructed with the edge-weighted Sankoff-Rousseau algorithm, indicating that all differences observed are either due to the edge lengths in the tree or the weights of specific adjacencies.

3.4 Discussion

In this chapter, we introduced the Small Parsimony Problem under the SCJ model with adjacency weights, together with an exact parameterized algorithm for the optimization and sampling version of the problem. The motivation for this problem is twofold: incorporating sequence signal from aDNA data when it is available, and recent works showing that the reconstruction of ancestral genomes through the independent analysis of adjacencies is an interesting approach [10,28,46,92].

Regarding the latter motivation, we address a general issue of these approaches that either ancestral marker orders are not consistent or are quite fragmented if the methods are constrained to ensure consistency. The main idea we introduce is to either take advantage of sampling approaches recently introduced in [28] or include available aDNA data to weight potential ancestral adjacencies and thus direct, through an appropriate objective function, the reconstruction of ancestral marker orders.

Our results on the mammalian data set suggest that this approach leads to a robust ancestral genome structure. However, we can observe a significant difference with a DCJ-based ancestral reconstruction, a phenomenon that deserves to be explored further. Our algorithm, which is based on the Sankoff-Rousseau algorithm similarly to

several recent ancestral reconstruction algorithms [10, 28, 92], has a fixed parameter tractable time complexity and can handle real instances containing a moderate level of syntenic conflict. Our experimental results suggest that introducing weights on adjacencies in the objective function has a significant impact in reducing the fragmentation of ancestral marker orders, even with an objective function with balanced contributions of the SCJ evolution and adjacency weights. For highly conflicting instances, it can be discussed if a reconstruction through under the assumption of parsimony is the right approach to solve these conflicts or if these should be addressed differently.

Our sampling algorithm improves on the Gibbs sampler introduced in [92] in terms of computational complexity and provides a useful tool to study ancestral genome reconstruction from a Bayesian perspective. Moreover, our algorithm is flexible regarding the potential ancestral marker adjacencies provided as input and could easily be associated with other ideas, such as intermediate genomes for example [47].

There are several further research questions opened by this work. From a theoretical point of view, we know the problem we introduced is tractable for $\alpha = 0$ and $\alpha = 1$, and it is shown to be hard for $\alpha > 33/34$ [79], but it remains to see whether it is hard otherwise. Further, given that the considered objective is a combination of two objectives to be optimized simultaneously, Pareto optimization is an interesting aspect to be considered.

Our model could also be extended towards other syntenic characters than adjacencies, i. e. groups of more than two markers, following the ancient gene clusters reconstruction approach introduced in [138]. As ancestral marker orders are defined by consistent sets of adjacencies, the principle of our dynamic programming algorithm could be conserved and it would be a matter of integrating gene clusters into the objective function, especially as conflicting instances are not as easy to define and compute as for adjacencies [105, 150].

From a more applied point of view, one would like to incorporate duplicated and deleted markers into the Small Parsimony Problem. There exist efficient algorithms for the case of a single adjacency [10, 28] that can provide adjacency weights, and natural extensions of the SCJ model to incorporate duplicated markers. However it remains to effectively combine these ideas.

Finally, again due to the flexibility and simplicity of the Sankoff-Rousseau dynamic programming algorithm, one could easily extend our method towards the inference of extant adjacencies if some extant genomes are provided in partially assembled form following the general approach described in [4, 6].

Chapter 4

Mind the gap: completing ancestral marker orders

In this chapter, we deal with the last step in scaffolding, which is filling the gaps between contigs ordered into scaffolds. The method was developed in the context of the analysis of two ancient *Yersinia pestis* data sets that was published as a preprint in [78] and will be described in more detail in Chapter 5, which is why the argumentation in this chapter is often based on *Yersinia pestis* as an example organism.

After reconstructing an order of markers using a local or global reconstruction approach, the markers still represent only a part of the actual genome. Gaps between contigs, that are the basis of markers in our setting, can correspond to for example repeat regions or less conserved regions that were not assembled, hence these regions are excluded when unique and/or universal marker families are computed. While short aDNA reads can be mapped onto one or several extant reference genomes to detect important evolutionary signals such as SNPs and small indels [113,127], analyzing the evolution of genome organization requires the assembly of the reads into longer contiguous sequences. However, highly fragmented assemblies make it challenging to exploit aDNA sequencing data for this task, including the analysis of important features such as the evolution of repeats and large scale genome rearrangement. Especially for repeat sequences, an initial de novo assembly needs to be improved by closing the gaps between ancient contigs in a scaffold.

The method FPSAC [117] aims to fill these gaps with putative sequences reconstructed from multiple sequence alignments of conserved extant genome regions. The method was applied to scaffold an ancient *Yersinia pestis* genome analyzed in Chapter 5, where gaps accounted for roughly 20% of the genome size of the ancestor, especially

taking advantage of the high sequence conservation in *Yersinia pestis* to reconstruct reliable gap sequences. The closing of the gaps shed an interesting light on genomic features hidden within the assembly gaps, in particular IS and their correlation with rearrangement breakpoints. However, the scaffolding of adjacencies and gap sequences obtained in [117] were inferred through computational methods within a parsimony framework, which can be sensitive to convergent evolution that cannot be ruled out for genomes with a high rate of genome rearrangements such as *Yersinia pestis* [36].

In traditional scaffolding experiments, if re-sequencing of these specific regions – e. g. using long-read sequencing technologies – is not feasible, gaps are usually closed using an estimate of the gap length through paired-end or mate-pair read mapping and then assembling potential reads for this gap locally, matching the expected length of the gap. Several assembly tools like ABySS [133] or Allpaths-LG [55] integrate methods for the gap filling, but also stand-alone tools [112, 121] have been developed. See [98] for detailed strategies that have been applied to finish several extant *Yersinia* draft genomes. In the context of aDNA reads however, such approaches are usually not successful as reads are too short and the read coverage is not sufficient (as will be shown in Chapter 5).

In this chapter, we introduce the method AGapEs (Ancestral Gap Estimation). For a potential adjacency between two ancient contigs, the method attempts to fill in the inter-contig gap sequence by selecting a set of overlapping aDNA reads that minimizes the edit distance to a template sequence obtained from the extant genome sequences that support the adjacency. This also allows us to pay special attention to specific annotations in the gap between two contigs, e. g. Insertion Sequences. In particular, when the presence of an IS is doubtful due to a mixed signal of presence/absence in the supporting extant genomes, a pair of templates can be considered, respectively including and excluding the IS.

In the following, we first introduce the general idea of the gap filling method AGapEs. Afterwards, we present a pipeline based on this approach to confirm sets of ancestral adjacencies by the available read data and also provide the opportunity to directly solve conflicting adjacency signals in reconstructions that are not yet consistent. It allows us to compare the consistent results of the methods presented in the previous chapters with a reconstruction that is mostly confirmed by the read data directly.

In Chapter 5, we apply this strategy to two data sets of aDNA reads for ancestors of the human pathogen *Yersinia pestis* [25, 91]. For both data sets, we obtain an assembly with reduced fragmentation and are able to fill a large number of inter-contig

gaps with aDNA reads. We identify several genome rearrangements between the ancient strains and extant *Yersinia pestis* genomes, however observe only a single small inversion between both ancient strains, suggesting that the genome organization of the agent of the second major plague pandemic was highly conserved.

4.1 Gap Filling as a shortest path problem

The input we take into account are the marker orders for extant genomes in a phylogeny and aDNA reads for the ancestral genome of interest. In the following, we describe a local strategy to fill gaps between two potentially adjacent markers in the ancestral genome.

Ancestral marker adjacencies For extant genomes, extant adjacencies can be observed directly, while for an ancestral genome of interest, we can determine a set of potential ancestral adjacencies. It can consist of e. g. all adjacencies observed in the extant genomes or based on the Dollo parsimony principle as used in [117]: two ancient marker extremities are potentially adjacent if there exist two extant genomes whose evolutionary path contains the ancestral genome of interest and where the two corresponding extant marker extremities are contiguous. This restricts the conflicts between potential ancestral marker adjacencies to conflicts between adjacencies conserved by the Dollo parsimony criterion. We will rely on this assignment of potential ancestral adjacencies in the next section.

Consequently every potential ancestral adjacency is supported by a set of extant adjacencies. A *gap* is the sequence between the two marker extremities defining an adjacency. Therefore each ancestral gap is likewise supported by a set of extant gap sequences. The key element of the approach we will describe lies in defining a *template* sequence or a set of alternative template sequences associated to each potential ancestral gap. We follow the general approach described in [117], that computes a multiple sequence alignment of the supporting extant sequence gaps and applies the Fitch parsimony algorithm [54] to each alignment column to reconstruct a most parsimonious ancestral sequence.

If the multiple sequence alignment of extant gaps shows little variation, as is the case for most gaps in our data sets, then a single template sequence can be considered, as we expect that minor variations compared to the true ancestral sequence (substitutions and small indels) will be corrected during the local assembly process. Alternatively, if larger variations are observed, such as larger indels or a contradicting pattern of

presence/absence of an IS in the supporting extant gaps, then alternative templates can be considered, under the hypothesis that the true variant can be recovered from the mapped aDNA reads.

4.1.1 Assembly of ancestral gap sequences from aDNA reads

We introduce a template-based method to assess the validity of an ancestral adjacency. The general principle is to associate to every ancestral gap a template sequence obtained from the supporting extant gaps sequences. We can then map aDNA reads onto this template and assemble the mapped reads into a sequence that minimizes the edit distance to the template sequence.

Definition 12 (Template-based Gap Filling Problem)**.** *Given a set of reads $\mathcal{R}$ and let d be an edit distance between two nucleotide sequences. For a potential adjacency between two oriented markers and a template gap sequence t, find a sequence of perfectly overlapping reads $\mathcal{R}_t \subseteq \mathcal{R}$ mapping to t that is minimizing*

$$\sum_{r \in \mathcal{R}_t} d(r, t).$$

The rationale for this template-based approach is that, due to the low coverage of the aDNA reads and their short length, existing gap-filling methods fail to fill a large number of ancestral gaps. For example, the method gap2Seq [121] is a recent efficient gap-closing algorithm based on finding a path of given length corresponding to the expected length of the gap in a de Bruijn graph. However the method is not able to fill roughly half of the ancestral gaps of an ancient DNA data set analyzed in Chapter 5 (see Table 5.4 on 88).

In the following, we will treat each potential ancestral adjacency separately, hence describing the details of our gap filling approach for a single ancestral adjacency.

Ancestral Gap Estimations (AGapEs) We will show how to find a solution to the Template-based Gap Filling Problem as a solution to a shortest path problem in a graph. Assume we are given a template sequence t for a gap in an adjacency between two oriented markers m_1 and m_2. We define $S = m_1 + t + m_2$ as the concatenated nucleotide sequence of the oriented markers and the respective template. We first align all aDNA reads onto S, e. g. using BWA [75], where we only consider mappings whose start and/or end position is in t, i. e. either fully included in t or overlapping

the junction between a marker and t. We denote the set of all such mappings as $\mathcal{M}_t$. The set of reads $\mathcal{R}$ defined by the mappings $\mathcal{M}_t$ is the input for the problem described in Definition 12, and our objective is to find a sequence of overlapping mappings that is covering the full gap and minimizes the edit distance to t.

When sorting all mappings $r \in \mathcal{M}_t$ by their start coordinates in S, we cut the first mapping such that its end coordinate corresponds to $|m_1|$. Taking this mapping as a start then assures that we have at least one overlapping pair of reads covering the junction between m_1 and t. Next, we construct a graph $G(V, E)$ where vertices are mappings $r \in \mathcal{M}_t$ and there is an edge between two vertices if the two mapping coordinates (segments of S) overlap. For each such edge $e \in E$, we define x_e as the non-overlapping suffix of the mappings with the highest end coordinate. We can associate a weight to each edge given by the edit distance between x_e and the substring S_x of S it aligns to.

A sequence of overlapping mappings that covers t with minimal distance can be found by searching for a shortest path in G between the vertex labeled with the smallest start position (i. e. the first mapping ending at the junction between m_1 and t) and the vertex labeled with the largest start position (i. e. the last mapping covering the junction between t and m_2). See Figure 4.1 for an illustration. If such a path exists, it can be found with Dijkstra's algorithm [41] implemented based on a min-priority queue in $O(|E| + |V| \log |V|)$ time. If no such path exists, then there are either regions in t that are not covered by any mapped aDNA read or mapping breakpoints, where two contiguous bases in the sequence are covered, but not both by the same read and hence no overlapping mappings can be found. In these cases, uncovered regions and breakpoints can be identified in the mappings beforehand to identify start and end vertices for several shortest paths to obtain a partial gap filling, precisely for the regions covered by mapped reads.

The sequence of read mappings can then be assembled into a corrected ancestral gap sequence while paying attention to the marker junctions covered by the last read. As we already know all the overlaps between reads based on their mappings, we only need to concatenate the suffixes x_e for each edge in the shortest path respectively, cutting all suffixes covering the marker sequences. Further, for each template that is only partly covered by mapped reads, we can correct the covered parts according to the read sequence and revert to the template sequence otherwise in a partial gap filling.

The same principle has been developed independently to correct sequencing errors in PacBio long reads with available short reads on the same dataset [59]. While the

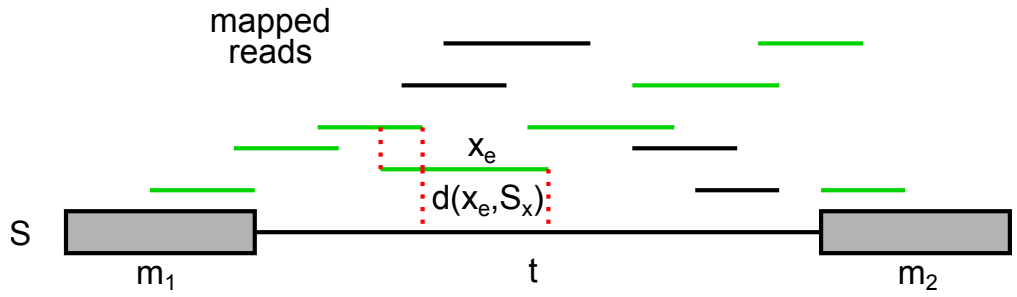


Figure 4.1: Example for a mapping of ancient reads to an adjacency in combination with a template gap sequence. We define a graph based on overlapping mappings of read to the adjacency and find a shortest path in this graph representing a sequence of reads that minimize the edit distance d to the template sequence.

long reads correspond to the template defined above, the short reads are expected to be more accurate and a mapping of these reads can hence be used to correct the long read. However the distance between the template and the mapped read sequence can be assumed to be smaller than in this application.

4.2 Local ancestral reconstruction based on Gap Filling

The AGapEs method can generally be applied as a follow-up step after a reconstruction method that produces a consistent order of markers for an ancestral genome of interest, e. g. as described in the previous two chapters. In addition, we can also use the illustrated gap filling strategy to avoid optimization decisions beforehand and directly start from a possibly conflicting set of adjacencies, for example defined with a simple parsimony criterion. In the following, we describe a local reconstruction pipeline that takes advantage of the available aDNA data as much as possible. Given a set of potential marker adjacencies that are possibly conflicting, we can solve conflicts directly based on the coverage in the aDNA data. In particular, we can include features of the genome sequence in the assembly gaps that are known to influence the rearrangement of genomes.

IS annotations Insertion sequence elements are simple transposable elements in bacteria that only encode the gene required for its own transposition [86, 130]. In *Yersinia pestis*, the expansion of four major IS families has been identified as the cause of gene loss during the emergence of *Yersinia pestis* from *Yersinia pseudotuberculosis* through gene inactivation at the sites of insertion. In addition, these highly transposable ele-

ments are known to be the cause of widespread genome rearrangements and hence an important key to analyze the rearrangement history of a species in a phylogeny. These repeated sequences however propose a challenge for de novo assembly [142], hence we include the coordinates of all IS annotations in all extant genomes to identify ancestral gaps that potentially contain one or several IS elements in the following.

4.2.1 Local reconstruction pipeline

We divide the set of potential ancestral adjacencies into three different groups: *simple*, *conflicting* and *IS-annotated* adjacencies.

An ancestral adjacency is *IS-annotated* if at least one of the supporting extant gaps is annotated as containing at least one IS. Due to the apparent high dynamic of IS, the presence or absence of the IS in the extant gaps pose the question if the IS element was also present in the ancestor.

In addition, these IS elements are also often present in adjacencies that are *conflicting* with other potential adjacencies, i. e. adjacencies sharing the same marker extremities. This implies the problem to decide which adjacencies to discard as a putative false positive to reconstruct a linear or circular ancestral genome structure. By mapping the available ancient reads to conflicting potential adjacencies, we can analyze which adjacency has support by the read data in terms of read coverage and hence a stronger signal to be ancestral. In order to obtain high confidence scaffolds for the ancestral genome, we will discard all conflicting adjacencies without support by the read data. Note that if a gap is conflicting and IS-annotated, we assign it to the conflicting group.

Finally, if an ancestral adjacency is not conflicting and none of the associated extant gaps is annotated with an IS, we call it *simple*. Simple adjacencies are the ones for which the extant support is robust and that we can hope to confirm using unassembled aDNA reads that align well on a reconstructed ancestral gap sequence obtained from a multiple sequence alignment of the supporting extant gaps [117].

The local reconstruction pipeline is summarized in Figure 4.2. For simple adjacencies, we can apply the AGapEs method described before directly. If the extant gap sequences are well conserved, we can compute a high quality template sequence as a basis for the read mapping and then correct the template according to the aDNA reads, depending on the mapping coverage that can be reached either completely or partially.

For conflicting adjacencies, we can use the gap filling analysis for all adjacencies in a conflicting component to identify mapping breakpoints or uncovered regions that

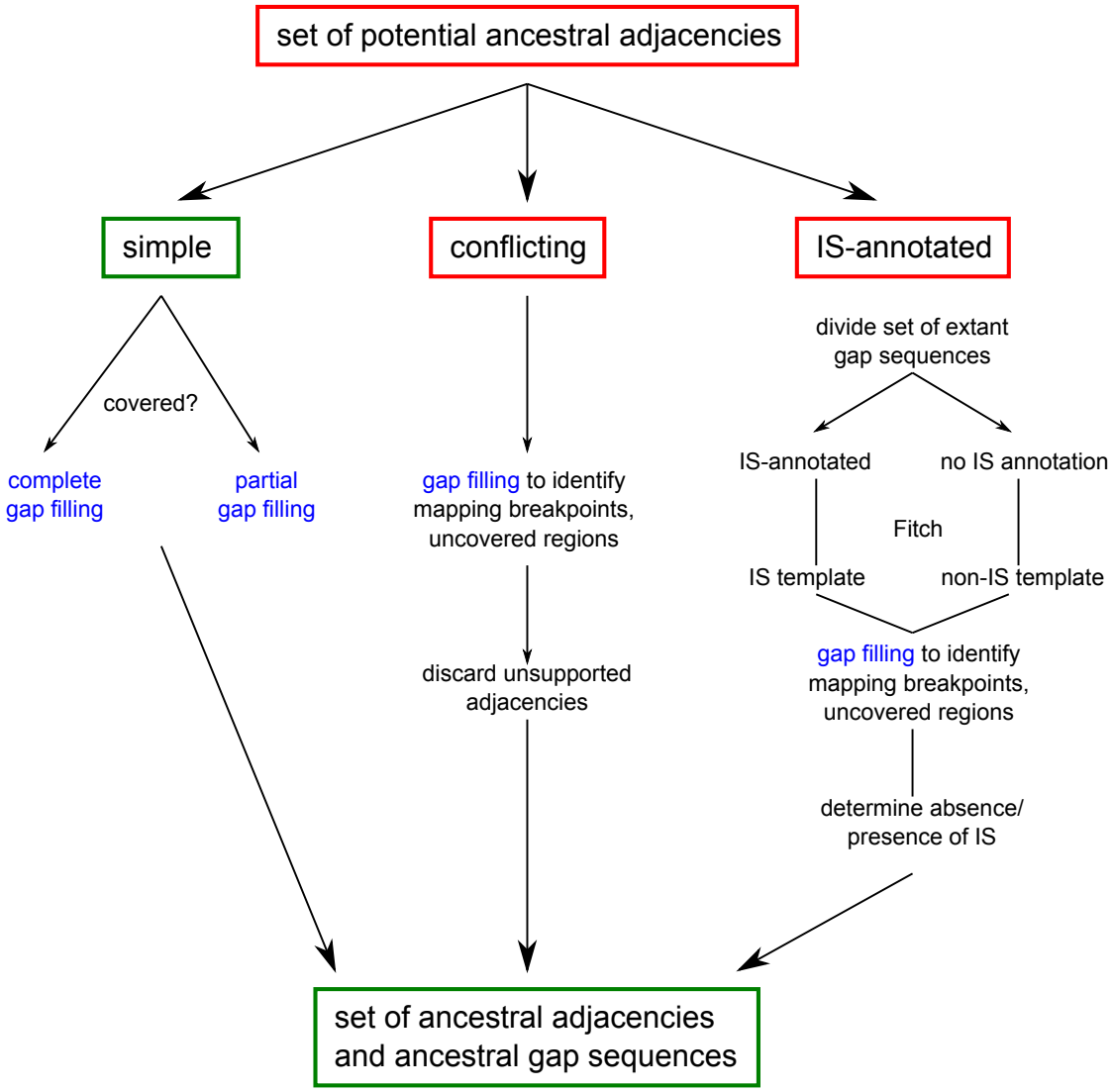


Figure 4.2: Local reconstruction pipeline based on filling the gaps between potential adjacencies with sequences of aDNA reads that minimize the distance to a template sequence. We differentiate between simple, conflicting and IS-annotated potential adjacencies respectively. For conflicting and IS-annotated adjacencies, we can use the AGapEs method to identify the supported ancestral adjacency and the supported variant of the ancestral gap sequence.

can indicate how to solve these conflicts. As no shortest path can be found in the mapping graph in these cases, the AGapEs method naturally identifies these features when attempting to fill the gap. Note that at this point, we solve the conflicts between potential adjacencies only based on the aDNA read data. One has to be careful to not interpret uncovered regions as missing data while they can also indicate false ancestral adjacencies. We discard any conflicting adjacency that is not sufficiently covered by aDNA reads, even if this removes complete conflicting components from the reconstruction.

For IS-annotated gaps, we divide its supporting extant gaps into sets of annotated and non-annotated extant sequences respectively and build a multiple alignment on each of these sets separately. This allows us to define two alternative templates with the Fitch algorithm that can be used as a basis to fill the gap, while breakpoints and uncovered regions in the gap variants identified by our method help to determine the supported variant for each IS-annotated gap.

4.3 Discussion

The AGapEs method presented in this chapter provides a useful way to fill inter-contig gaps in ancient reconstructed genomes where classical gap filling methods cannot be applied. It relies on a template sequence e. g. parsimoniously reconstructed from supporting extant gap sequences and utilizes unassembled aDNA reads to fill the gap locally. The strategy has been extended to a local reconstruction strategy confirming ancestral adjacencies with aDNA support and indicating all adjacencies that are not covered by reads.

The method relies on the quality of the aDNA data, as missing data does not allow to fill a gap completely. However the relation to extant genomes still provides a point of reference for regions of the ancient genome that cannot be retrieved due to degradation of the DNA material. Also, regions with very low coverage involve the danger of erroneously "correcting" the template sequence due to sequencing errors or aDNA damage apparent in the reads. If there is no support for all adjacencies in a conflicting component, it will result in a more fragmented solution in comparison to optimization methods. All these points reveal critical issues for methods that rely on aDNA data as the main source of information, which need to be taken into consideration in the analysis of these data. However they also provide the opportunity to point out weak adjacencies in reconstructions by optimization based methods.

For aDNA data with sufficient quality in terms of read coverage however, the combination of the local assembly of reads and parsimonious gap sequence reconstruction can improve the number of closed gaps considerably, as can be seen in our analysis of two ancient *Yersinia pestis* genomes in the following chapter. With the on-going improvement of aDNA sequencing methods, it provides an opportunity to de novo assemble the full nucleotide sequence of ancient genomes.

Chapter 5

Reconstruction and analysis of two ancient *Yersinia pestis* strains

This chapter describes the reconstruction of genomes in the *Yersinia pestis* phylogeny, including two sequencing data sets of ancient *Yersinia pestis* strains. We will first start with a detailed local analysis of both ancient data sets separately using the gap filling approach described in Chapter 4. We then apply the global methods presented in Chapters 2 and 3 to one of the data sets and compare the result to the reconstruction obtained with the local method FPSAC [117]. Finally, we combine the information from both ancient data sets in a global reconstruction as described in Chapter 3, and compare the proposed reconstructions by all methods for both ancient *Yersinia pestis* genomes.

5.1 Sequencing data and reference genomes

The first aDNA data set was obtained from the remains of a London victim of the Black Death pandemic in the 14th century [19], the second consists of five samples from victims of the Great Plague of Marseille around 400 years later [18]. We will refer to them as the *London* and *Marseille* data sets in this chapter respectively.

The read set for the London strain (Genbank accession SRA045745) consists of merged single-end reads sequenced from dental material found on a excavated medieval cemetery in London. The sequencing material was obtained by array-based enrichment using the extant *Yersinia pestis* strain CO92, and the enriched libraries were sequenced on the Illumina Genome Analyzer IIx platform [19]. We used the sequencing data for individual 8291 in this analysis. The average read length is 53 bp

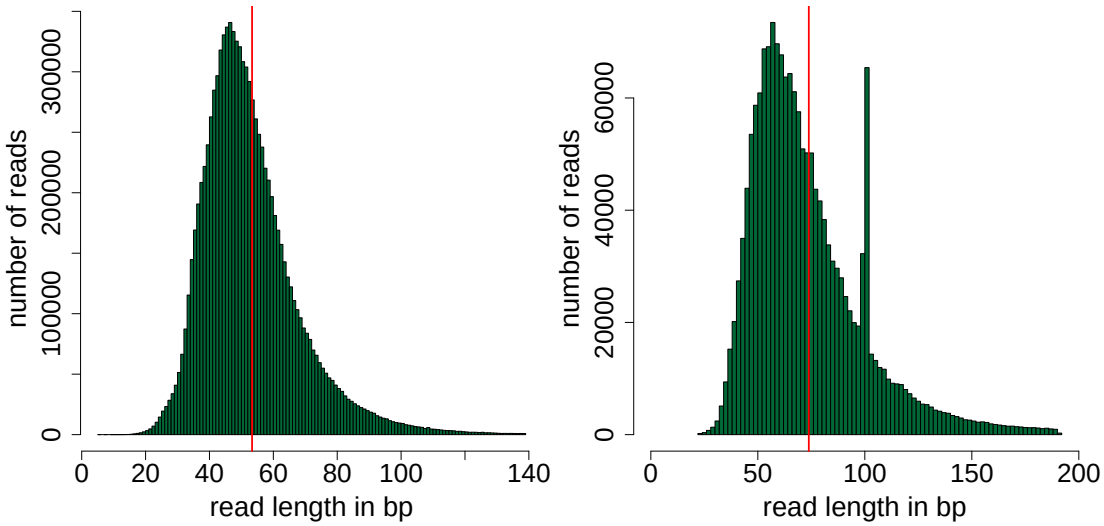


Figure 5.1: Read length distribution of individual 8291 in the London data set (left) and Marseille sample OBS116 (right) after preprocessing. The mean read length in each data set is indicated in red.

in this data set (see Figure 5.1), with a coverage estimate of 28.2 reads per site for the chromosome.

The read set for the Marseille strain (ENA accession PRJEB12163) contains five samples from victims of the Great Plague of Marseille in 1722, obtained again by array-based enrichment using strain CO92 and additional chromosomal regions from other *Yersinia pestis* strains that are absent in CO92. Enriched libraries were sequenced on an Illumina HiSeq 2000 [18]. For this data set, we preprocessed the reads as partly described in [18] by first trimming adapters separately for both paired-ends using cutadapt [90] for the adapter sequence AGATCGGAAGAGC, a maximum allowed error rate of 0.16 and a minimum overlap length between read and adapter of 1. We merged paired-end reads with negative insert size with flash [85] with a minimum required overlap length of 11 and finally filtered all reads shorter than 24 bp. The average read length is 75 bp in the five Marseille samples (see Figure 5.1).

In all analyses, we rely on seven extant *Yersinia pestis* and four *Yersinia pseudotuberculosis* as reference genomes. For all these references, a fully assembled genome sequence is available, the respective accession numbers are given in Table 5.1. As the ancient data does not contain sequencing information about the plasmids, we restrict the analysis to the main chromosome of the bacteria in the following.

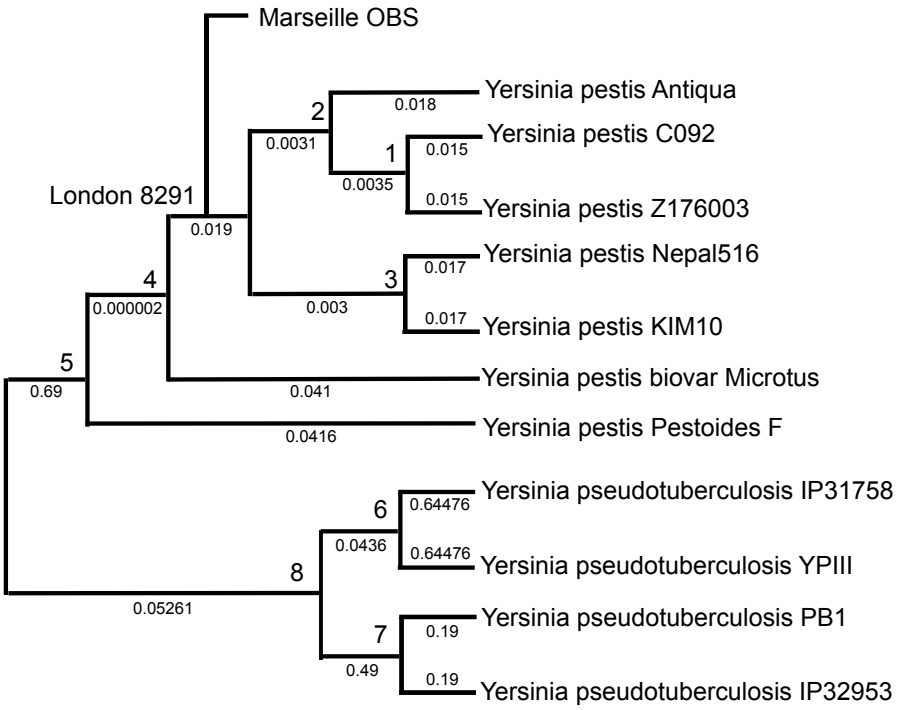


Figure 5.2: Underlying phylogeny of extant *Yersinia pestis* and *Yersinia pseudotuberculosis* species used as reference genomes and the location of both considered ancient *Yersinia pestis* data sets.

Table 5.1: Accession numbers of full assemblies for the chromosome of all extant *Yersinia pestis* and *Yersinia pseudotuberculosis* reference genomes. In addition, the number of IS annotations per reference is given.

strain	accession no.	IS annotations
Yersinia pestis		
CO92	NC_003143.1	233
Antiqua	NC_008150.1	293
Z176003	NC_014029.1	170
Nepal516	NC_008149.1	212
KIM10+	NC_004088.1	151
biovar Microtus 91001	NC_005810.1	168
Pestoides F	NC_009381.1	190
Yersinia pseudotuberculosis		
IP 31758	NC_009708.1	-
YPIII	NC_010465.1	-
PB1/+	NC_010634.1	-
IP32953	NC_006155.1	-

The phylogeny of the considered strains is depicted in Figure 5.2 and is taken from [18, 19], inferred from mutation analysis. The London strain is assumed to be ancestral to five extant *Yersinia pestis* genomes. In addition, the ancient Marseille strain is assumed to be a direct descendant of the London strain, but not ancestral to any sequenced extant strains. Hence it is placed as an extinct leaf in the phylogeny.

Insertion sequence annotation. Since Insertion Sequences (IS) are strongly related to rearrangements in *Yersinia pestis* evolution, their annotation in the considered extant genomes is crucial. In order to complete IS annotations in the reference genomes, the following annotation pipeline has been designed in [78]. First, already annotated IS were extracted for all extant *Yersinia pestis* genomes from their Genbank files. Then, they were completed by an automatic annotation using the Basys annotation server [144] and Hidden Markov Models (HMM) trained for 11 different IS families, implemented using `hmmer` [44]. The number of IS annotations per reference genome ranges from 151 in *Yersinia pestis KIM10+* to 293 in *Yersinia pestis Antiqua* (see Table 5.1). The length of the annotations ranges from 60 bp to 2,417 bp as can be seen in Figure 5.3. We see several annotations of similar length arising from IS families with a high frequency. Some short annotations deviate from the expected length for IS of at least 500 bp. They can indicate incomplete IS annotations and are frequently overlapping with additional, longer annotations. However, in order to avoid filtering any true annotations, we include them all as potential IS coordinates in the following analysis.

5.2 Local reconstruction of both ancient strains

Applying our reconstruction pipeline described in Chapter 4 to both ancient *Yersinia pestis* data sets, we first show the consistency of genome organization in reconstructions based on a de novo and a reference-based assembly for the London outbreak strain. With the reconstruction of the Marseille outbreak strain, we will then analyze the genome evolution between both ancient genomes and extant genomes in terms of genome rearrangements. This analysis has been published as a preprint in [78].

5.2.1 Reconstructing the London outbreak strain

In order to assess the impact of the initial contig assembly on the final result, we considered two contig assemblies of the aDNA reads. Bos et al. [19] describe a reference-

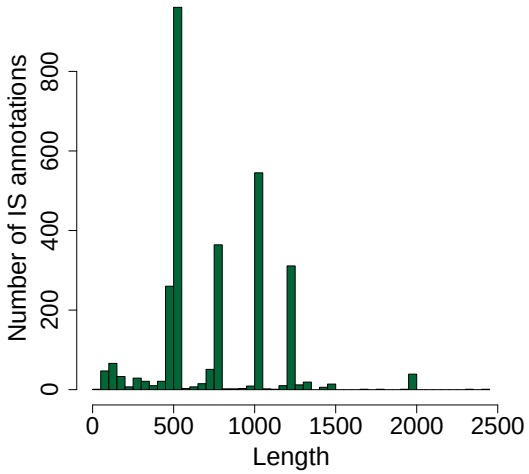


Figure 5.3: Lengths of all potential IS annotations in all *Yersinia pestis* reference genomes.

based assembly of the London strain consisting of 2,134 contigs of length at least 500 bp that add up to a total length of 4,013,159 bp. It was obtained with the assembler Velvet [153] again using the extant strain *Yersinia pestis* CO92 as a reference. In order to assess the influence of the reference sequence on the assembly, we additionally de novo assembled the ancient DNA reads into contigs using Minia [30]. Minia is a conservative assembler based on an efficient implementation of the de Bruijn graph methodology. In general, the tool produces shorter contigs, as it avoids assembly decisions in case of ambiguity in the sequence data. We assembled aDNA reads with Minia with different values of the k-mer threshold $k \in \{17, 19, 21\}$ and a minimal k-mer occurrence of 3. We evaluated the total contigs length with regards to a minimal contig length threshold $\in \{200, 300, 400, 500\}$. The total contig length can indicate how much of the expected genome size the assembled contigs can cover, while a higher minimal contig threshold can provide a better base for defining markers. We found the best trade-off with $k = 19$ and a minimal contig length of $300bp$ for the de novo assembly of the London data set. As expected, the de novo assembly is more fragmented with 4,183 contigs that cover 2,631,422 bp (see Table 5.2). We will refer to the assembly by Bos et al. as *reference-based* and the Minia assembly as *de novo* assembly in the following.

We first compared both assemblies by aligning them with MUMmer [73]. Most contigs can at least partly be aligned to a contig from the respective other set. The de novo

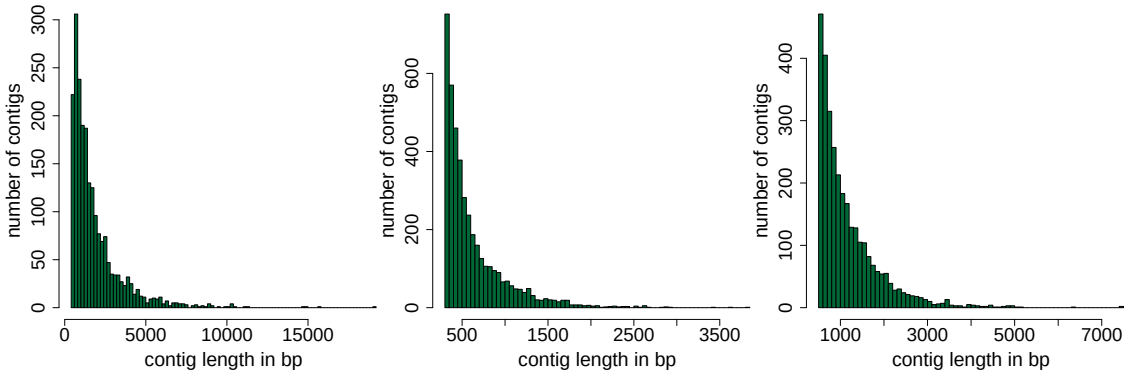


Figure 5.4: Contig length distribution for the reference-based assembly of the London data set (left), and both de novo assemblies of the London (middle) and Marseille sample OBS116 (right) respectively.

assembly however covers only 60% of the reference-based assembly. Unaligned bases mostly belong to regions in the reference-based assembly that have not been assembled in the conservative de novo assembly, and only an extremely low amount of nucleotide variations can be observed (Table 5.2), together with no observed genome rearrangement evident from the contigs. We collected the bases of reads mapping to these variable positions, but could not find any indication that the nucleotide variations are caused by the reference genome used in the reference-based assembly.

To allow the comparison with extant genomes, contigs above a minimum length threshold were aligned to the extant genomes to define families of markers as de-

Table 5.2: Comparison of sets of contigs obtained in reference-based assembly and de novo assembly for the London strain.

	reference-based assembly Velvet [19, 153]	de novo assembly Minia [30]
Length threshold L	500 bp	300 bp
Number of contigs $> L$	2,134	4,183
Total contig length	4,013,159 bp	2,631,422 bp
Aligned contigs	1,866 (87.44%)	3,885 (92.88%)
Aligned bases	2,414,881 (60.17%)	2,380,757 (90.47%)
Single Nucleotide InDels	14	
SNPs	39	

scribed in [117]. Marker families were filtered to retain only unique and universal families, i. e. families that contain exactly one marker in each considered genome (including ancient genomes). Subsequently, for the reference-based assembly, we obtained 2,207 markers that cover 3,463,281 bp in total. For the de novo assembly, we obtained 3,691 markers covering 2,215,596 bp in total. All markers of the de novo assembly are contained in or overlapping with markers from the reference-based assembly, with small non-overlapping regions in the de novo assembly marker set that are due to the segmentation process.

Reconstructing potential ancestral adjacencies. For the reference-based assembly, we inferred 2,208 potential ancestral adjacencies: 1,991 are simple, 207 are IS-annotated but not conflicting, and 10 are conflicting. Among the conflicting adjacencies 8 are also IS-annotated, illustrating that most rearrangements in *Yersinia pestis* that can create ambiguous signals for comparative scaffolding are associated with IS elements. For the de novo assembly, we obtain 3,691 potential ancestral adjacencies: 3,483 are simple, 201 are IS-annotated and non-conflicting, and only 7 are conflicting, including 5 IS-annotated adjacencies (see also Figure 5.6). The difference in the number of IS-annotated adjacencies between the two assemblies can be explained by larger gaps in the de novo assembly that contain potentially more than one IS sequence but are separated into several gaps in the reference-based assembly.

For most potential ancestral adjacencies, the lengths of the sequences in extant genomes associated with the supporting extant adjacencies are very similar, indicating well conserved extant gaps (Figure 5.5). We have 28 and 21 gaps in the reference-based

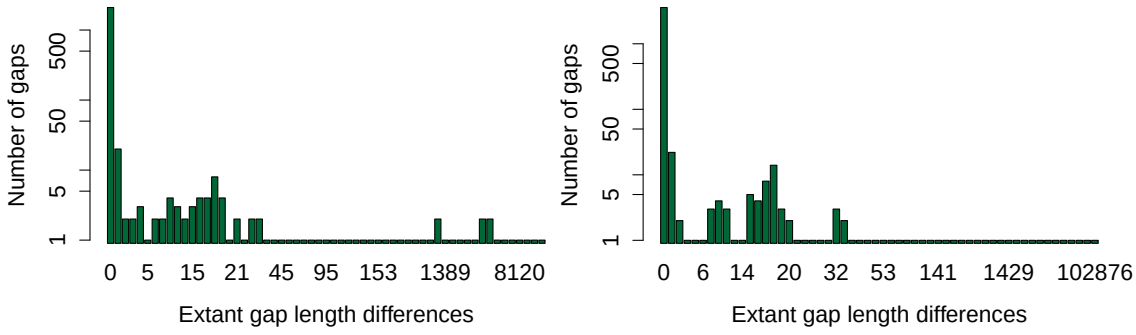


Figure 5.5: Differences in extant gap lengths for all markers of the London data set (left) and the Marseille data set (right).

and de novo assembly respectively whose length difference falls into the length range of potential annotated IS elements, thus raising the question of the presence of an IS within these adjacencies in the ancestral genome. We note a small number of potential ancestral adjacencies with strikingly large extant gap length differences (7 and 5 respectively) in the order of Kilobases. All of these gaps accumulate more than one IS annotation in some extant genomes. Most problematic are two gaps with length differences of more than 100 kbp. As these gaps are not well conserved in general (apart from the inserted sequences), it is difficult to obtain a good template sequence based on a very fragmented multiple alignment at this point. We will get back to these special gaps in the next paragraphs.

Ancestral gap filling We apply AGapEs to close all potential ancestral gaps. For mapping the reads to the template gap sequences, we used *BWA* [75] with parameter -a to keep all alignments for each read and samtools rmdup [76] to remove PCR-duplicates. In order to correctly identify breakpoints in the mappings, we also removed all clipped alignments.

We assume a gap to be filled, if we find a sequence of reads that covers the whole ancestral gap. As we test two alternative templates for an IS-annotated gap, we consider it filled if only one alternative is covered or if both templates are covered but the IS is only annotated in a single extant genome. In the latter case, we expect the non-IS gap version to be ancestral, as the IS was most likely obtained along the edge to the annotated extant genome. If otherwise both alternatives are covered, we cannot unambiguously recover the supported gap variant at this point and mark it as not filled. If a gap template sequence is only partially covered by mapped aDNA reads, we correct the covered regions as described above and use the template sequence of the uncovered regions to complete closing the gap. In Figure 5.6, the gap-filling results are summarized and detailed numbers are given in Table 5.3.

For both assemblies, a high number of gaps is supported by sufficient read coverage that enables us to fill the gap with a sequence of overlapping aDNA reads. Especially considering partially covered gaps for the de novo assembly improves the length of the genome that is supported by reads. We also find covering reads for all gaps of length 0, spanning the breakpoint between directly adjacent markers and hence confirming the adjacencies respectively. Gaps that are not covered by any reads and hence not filled indicate either genome regions that have not been sequenced or where a contradicting signal of presence or absence of an IS element could not be solved based on the reads.

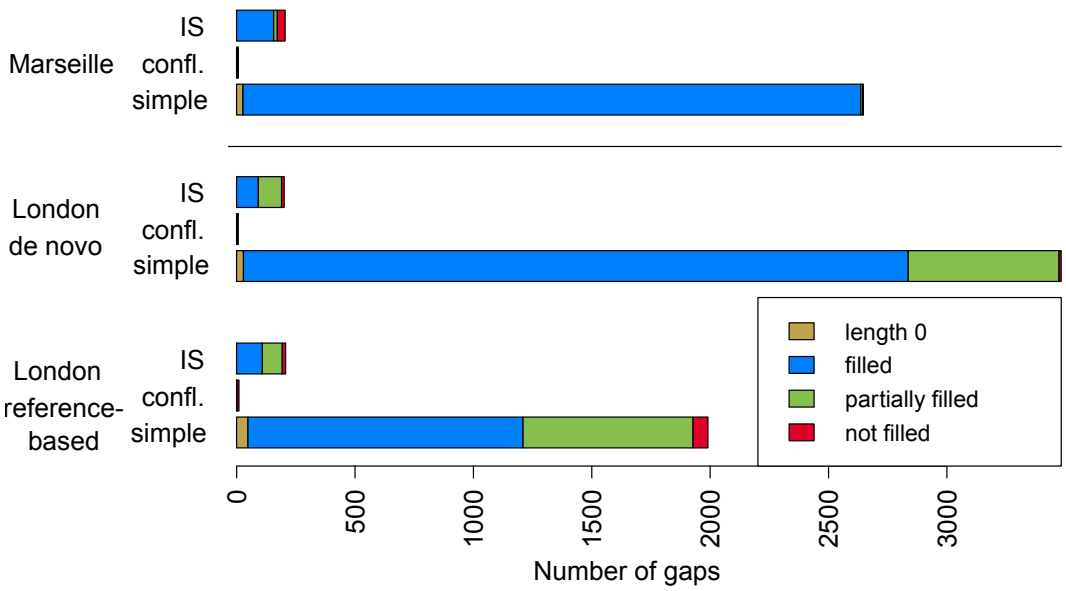


Figure 5.6: Results of gap filling for both London assemblies and the Marseille assembly. Note that if a gap is conflicting and IS-annotated, we assign it to the conflicting group. We differentiate between gaps of length 0 (i. e. both markers are directly adjacent), completely, partially and not filled gaps.

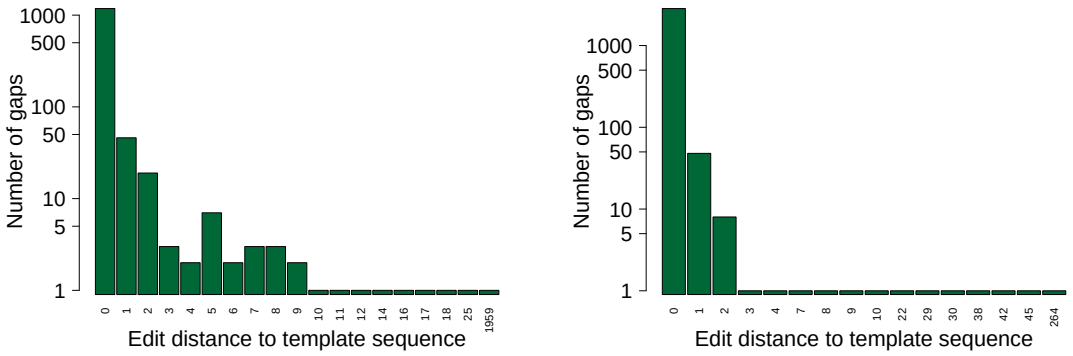


Figure 5.7: Edit distance between reconstructed gap sequence and template sequence for all fully covered gaps in the reconstruction for the reference-based assembly (left) and the de novo assembly (right).

Table 5.3: Detailed results of gap filling for both assemblies for the London data set and the de novo assembly for the Marseille data set. Note that if a gap is conflicting and IS-annotated, we assign it to the conflicting group. The length associated to gap regions uncovered by mapped aDNA reads is the length of the corresponding template sequence.

	London reference-based assembly			London de novo assembly			Marseille de novo assembly		
	simple	conflicting	IS	simple	conflicting	IS	simple	conflicting	IS
gaps of length 0		48			29			27	
gaps filled	1,162	2	109	2808	2	92	2610	4	157
length (bp)	172,614	7,876	70,550	710,138		86,805	751634	13231	222079
gaps partially filled	718	-	84	637	-	98	9	-	15
total length (bp)	319,633	-	240,085	862,307	-	505,856	15001	-	34223
covered by reads (bp)	245,779	-	194,414	765,406	-	443,090	6140	-	28650
gaps not filled	63	8	14	9	5	11	1	3	33
length (bp)	7,154		172,689	25,777		18,249	130		77125
total number of gaps	1,943	10	207	3454	7	201	2620	7	205
total assembly length	4,398,214			4,441,004			4,342,298		
coverage by marker	3,463,281 (78,74 %)			2,215,596 (49.88 %)			3,143,627 (72.40 %)		
coverage by reads	4,154,514 (94.46 %)			4,230,162 (95.25 %)			4,165,361 (95.93 %)		

We further computed the edit distance between the reconstructed gap sequence and the previous gap template (see Figure 5.7). For IS-annotated gaps, we computed the distance to a template sequence based on all extant gap occurrences, i. e. without considering the alternative templates as described previously. This allows us to compare the filled gap sequence with the reconstructed gap sequence if IS annotations are ignored. While the distance between reconstruction and template is very small for most gaps, we identified one case where the parsimonious gap sequence based on all extant occurrences of the adjacency excludes the IS element. The gap has a larger distance of 1959 to the template, corresponding to the annotated length of the IS for this gap. However if aDNA reads are mapped separately to alternative templates based on IS and non-IS annotated extant gaps, only the IS-annotated gap template is covered. The mappings of the reads shows clear breakpoints at the respective gap for the non-IS template and provides full coverage for the IS template.

For IS-annotated gaps, in both assemblies 95 ancestral gaps are reconstructed containing the IS, while 112 resp. 106 ancestral gaps are reconstructed without the IS. From the 95 IS gaps, 22 contain annotations that are shorter than 400 bp, however they all contain additional longer annotations in the same gap, confirming their classification as IS gaps and indicating that the short annotations are potentially incomplete. Analyzing the number of ancestral IS with a Dollo parsimony criterion considering only the extant IS annotations, we have 96 ancestral gaps that contain an IS, indicating a large agreement between the IS that are conserved by the parsimony criterion and the IS supported by aDNA reads.

Comparison with gap2Seq The gap2Seq algorithm aims at closing gaps in assemblies as an exact path length problem on a de Bruijn graph of the given reads. We ran gap2Seq on the reference-based assembly gaps with $k = 19$. For the de novo assembly gaps, we could only get results for a higher $k = 23$, while the implementation could not finish for lower values of k. In comparison, we can fill more gaps than gap2Seq when taking advantage of a template sequence in AGapEs, as detailed in Table 5.4.

Conflicting adjacencies Conflicting adjacencies are related by the marker extremities they share, defining clusters of related conflicting adjacencies. For the reference-based assembly, we identified three such clusters (see Figure 5.8). Two of them consist of three adjacencies that are all annotated with IS elements, while the other consists of four adjacencies, including two IS-annotated adjacencies. In total, only two of these conflicting adjacencies are supported by aDNA reads. All other adjacencies contain

uncovered regions indicating potential breakpoints. So in order to propose a conflict-free scaffolding, we chose to remove all unsupported conflicting adjacencies. Filling these gaps only partially does not provide much information, as uncovered regions can be either breakpoints or not sequenced regions of the ancestral genome.

For the de novo assembly, there are only two clusters of conflicting adjacencies that match with the clusters observed in the reference-based assembly according to

Table 5.4: Comparison of gap filling results for AGapEs and gap2Seq on the London data set. For each assembly, we divide all gaps into the three respective categories. We count gaps that are filled by both methods and gaps that are only filled by one of both methods. The total value sums up the number of gaps filled by each method.

	reference-based				de novo assembly			
	all	AGapEs	gap2Seq	both	all	AGapEs	gap2Seq	both
simple	1991	263	3	924	3483	1919	0	886
conflicting	10	3	0	0	7	3	0	0
IS	207	70	0	62	201	76	0	37
total	2208	1322	989		3691	2921	923	
		59,87%	44,79%			79,14%	25,01%	

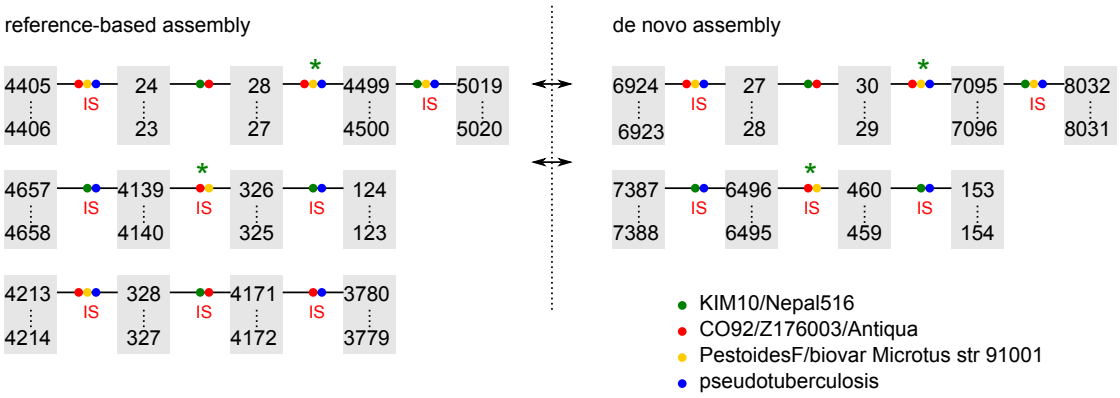
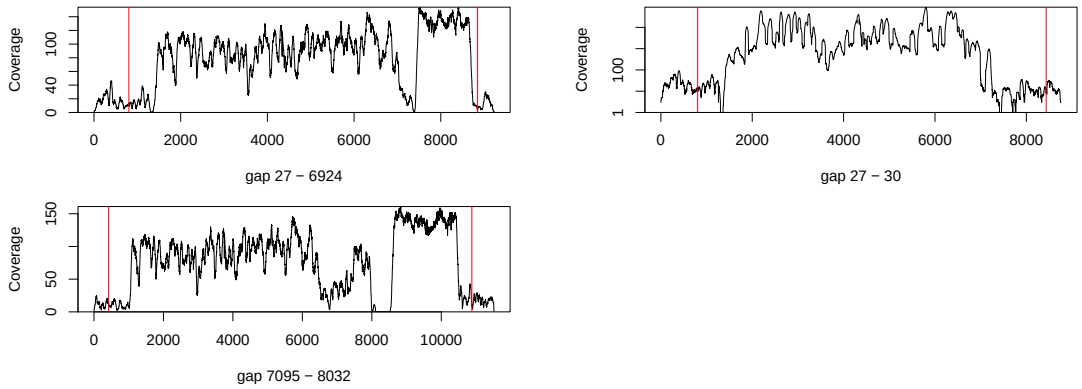


Figure 5.8: Conflicting components in the set of potential adjacencies of the London reference-based assembly and the de novo assembly. Markers are indicated by the grey boxes. Each marker x is represented by its extremities with $2x$ for the head and $2x - 1$ for the tail of the marker. The depicted IDs correspond to the marker IDs assigned in our data analysis. Adjacencies are depicted by connecting lines between two extremities. Gaps containing IS sequences are labeled accordingly. The color labels assigned to each adjacency indicate the extant occurrences and hence the conservation of the adjacency in the tree. All gaps that are fully covered by reads and do not contain breakpoints in the mappings are marked by green stars.

Component 1:



Component 2:

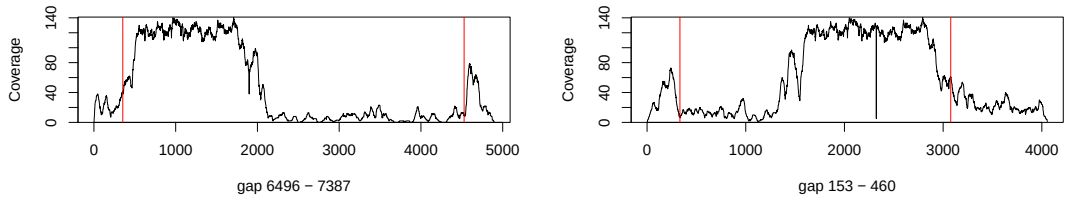


Figure 5.9: Read coverage for discarded adjacencies in conflicting components for the de novo reconstruction for the London data set. The gap sequence is flanked by the marker, the gap borders are indicated in red. All gaps have regions with no read coverage.

the coordinates of the supporting extant gaps. As the same adjacencies are covered by aDNA reads, we resolve the scaffolding conflicts identically to the reference-based assembly by keeping the two supported adjacencies and removing all other conflicting adjacencies. The read coverage of discarded adjacencies is shown in Figure 5.9.

For the reference-based assembly, the set of ancestral adjacencies can then be ordered into seven CARs, while we obtain five CARs for the de novo assembly. We convert the reconstructed sequences of markers back to genome sequences by filling the gaps with the read sequences if possible and resorting to the template sequence otherwise.

As mentioned earlier, we observe two corresponding gaps in both reconstructions respectively with highly differing extant gap lengths ($\geq 100,000$ bp) and very little conservation. While the extant gap coordinates are similar for both gaps, the multiple alignment of the extant gap sequences is in both cases very fragmented and hence the resulting template sequences are dissimilar, even though they are based on mainly the

same extant gap sequences. The mapping of reads onto these templates is poor: in the de novo assembly, the gap contains 211 uncovered regions of 9319 bp in total. An overview of the uneven read coverage for this gap in the de novo assembly is given in Figure 5.10. As the reconstructed sequences have a high edit distance after partial gap filling to each other, we cannot reconstruct a coinciding sequence in both reconstructions. Hence we remove these gap sequences completely from the reconstruction to avoid faulty reconstructed sequences.

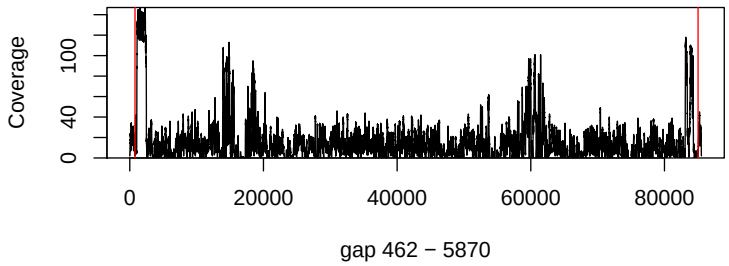


Figure 5.10: Large gap in the reconstruction of the London strain that has been removed from both assemblies due to insufficient read coverage.

Comparing the two improved assemblies To evaluate the impact of the initial assembly on the final result, we compared the two sets of CARs obtained from both initial assemblies by aligning the resulting genome sequences again using MUMmer [73]. As can be seen in Figure 5.11, we observe no rearrangements between both resulting sets of CARs, showing that the final result does not depend on the initial contig assembly in terms of large-scale genome organization.

We achieve a high similarity between both sets of CARs. While the improved de novo assembly contains a larger amount of filled gap sequences, we align nearly all of both sequences and observe only a low number of SNPs, insertions and deletions between both assemblies (see Figure 5.11 and Table 5.5).

The differences found are often located in gaps with low read coverage regions. If short regions in the gaps are only covered by a single read, in order to find a shortest path in the mappings, this read has to be included at all costs and can cause corrections to the template that are not supported by any other read. Further re-sequencing of these regions could clarify which variant is present in the ancient genome.

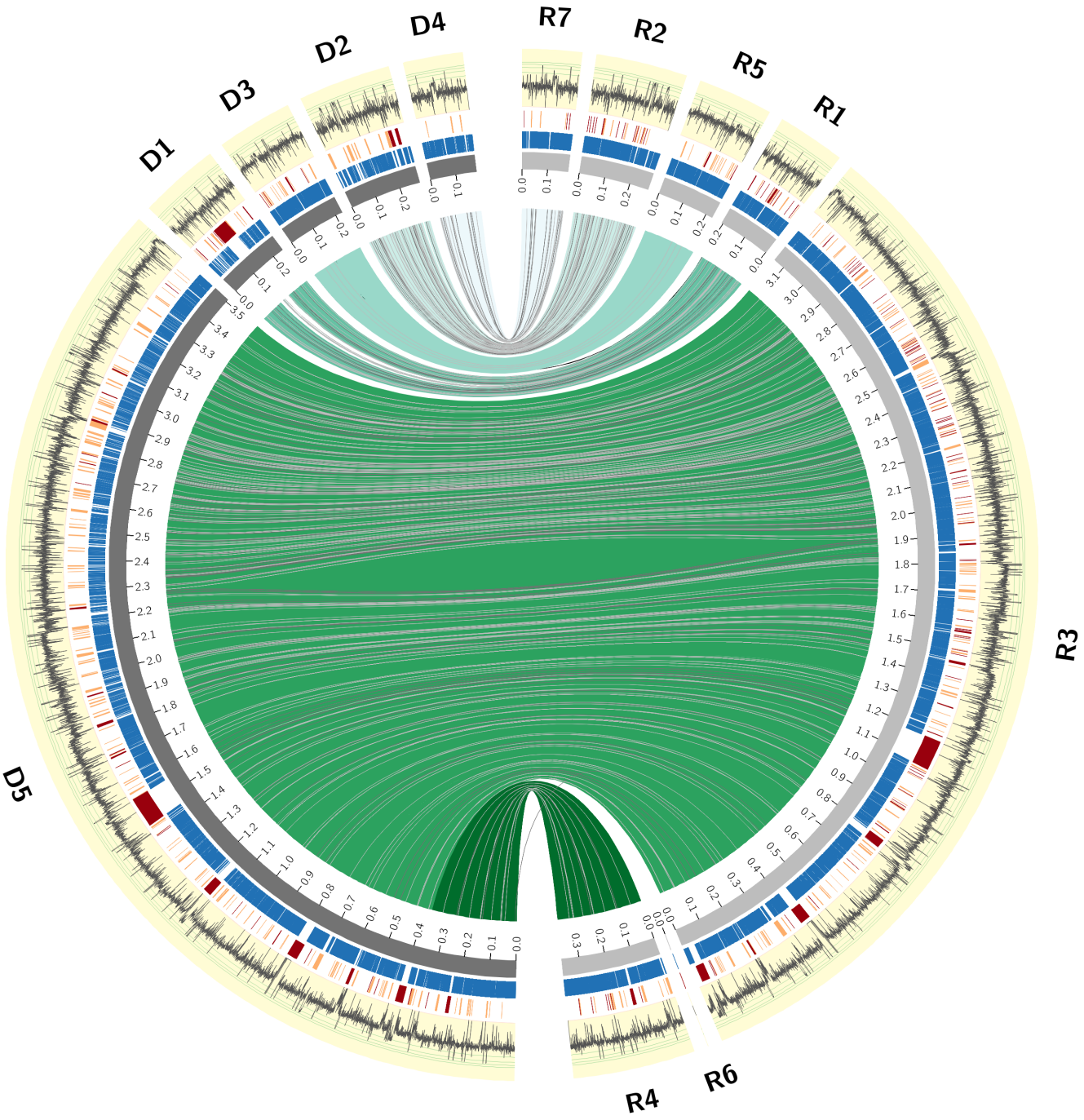


Figure 5.11: Comparison of the de novo assembly (left) and the reference-based assembly (right). The inner links connect corresponding CARs in the reconstructions. The grey lines indicate substitutions and InDels observed. The positions in both assemblies covered by markers are indicated in blue. All gaps that have IS annotation in the extant genomes are shown in orange. In addition, gaps that are only partially filled or have unconserved extant gap lengths are indicated in red in the same circle. Finally, the outmost ring shows the average read coverage in windows of length 200 bp in log-scale. The figure has been compiled with Circos [72].

In addition, we aligned all reads again to the final assembly to assess the amount of uncovered regions in the reconstructed sequences. In total, only 85,578 bp in the reference-based assembly and 88,529 bp in the de novo assembly are not covered by any read and most uncovered regions are rather short (see Figures 5.11 and 5.12). Based on these mappings, we ran the assembly polishing tool Pilon [147] on the final assembly. The tool identified several positions where the assembled base (also present in the template) is the minority in comparison to all reads mapping at this position. As Pilon is not taking the respective bases of the extant genomes into account, it runs the risk of correcting the assembly according to sequencing errors in the reads. In fact, the most frequent proposed substitutions correspond to the common damage pattern of cytosine deamination observed in aDNA [108]. As a consequence, we only kept small indel corrections but reject all single-base corrections.

Given the differing quality of the two considered assemblies, the resulting improved assemblies have a different ratio of subsequences defined by markers and gap sequences. In the reference-based assembly, 78.74% of the resulting sequence is defined by markers and hence directly adopted from the initial assembly, while for the de novo assembly only 49.88% of the assembly is based on marker sequences and a larger part is based on the filled gap sequences. Together with the gaps that have been filled by read sequences, we can say that for the reference-based assembly in total 94.46% and for the de novo assembly in total 95.25% are reconstructed using only the available aDNA reads, while the rest of the assembly is based on extant sequence information.

5.2.2 Reconstructing the Marseille outbreak strain

This data set consists of five samples as described in [18] that we assembled separately with Minia and parameter $k = 21$ as the k-mer length used to build the de Bruijn graph. Unlike for the London data set, there was no available reference-based assembly, and

Table 5.5: Comparison of improved assemblies on nucleotide level. Both sets of CARs have been corrected with Pilon [147], but only corrections of small Indels are kept.

	reference-based	de novo assembly
Aligned CARs	6 (85.71%)	5 (100%)
Total bases	4398441	4441094
Unaligned bases	13145(0.30%)	38702(0.87%)
Indels	216	
Substitutions	389	

a de novo assembly does not pose the risk of influencing the assembly by the choice of the reference sequence.

We first compared the quality of the five resulting assemblies by mapping contigs with a minimal length to the genome of the extant strain *Yersinia pestis* CO92 and summing the total length of the mappings as seen in Figure 5.13. While restricting the minimal contig length, two of the samples cover an extensively larger part of the reference and thus indicate a better sequencing quality. The differing quality of the samples can also be observed in Figure 5.15, showing the specific mapping positions of contigs to the reference strain. The notable deletion in all samples in comparison to *Yersinia pestis* CO92 has already been described in [18] based on a mapping of reads to the reference.

If we restrict the minimal contig length, only a small part of the reference genomes are covered by contigs from all five samples as shown in Figure 5.14. We conclude that the different samples should be pooled in order to achieve a good coverage of the

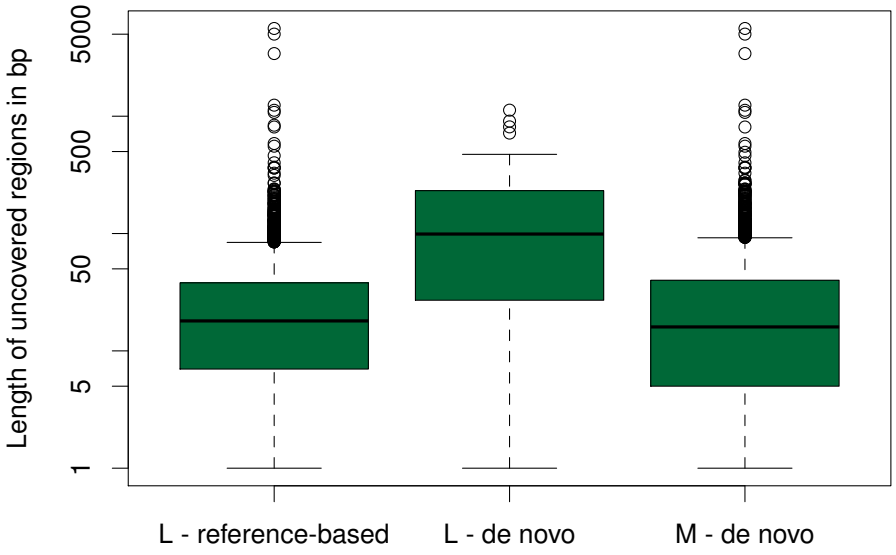


Figure 5.12: Length distribution of uncovered parts in bp after mapping all reads from the London and Marseille data sets back to the improved reconstructed sequence for each assembly respectively.

complete ancient genome, under the assumption that all samples represent the same ancient *Yersinia pestis* strain. In order to confirm this assumption, we compared all assemblies on the level of the contigs by alignment, but found no breakpoints between the assemblies of different samples that could indicate different underlying strains on the basis of the assembly.

Nevertheless, we rely on the assembly of sample *OBS116* with a minimal contig length of 500 bp to segment the extant genomes into markers. The assembly consists of 3,089 contigs with a total length of 3,636,663 bp. We also computed an assembly of the pooled reads from all samples that could however not extensively improve the quality of the assembly, so we only joined all sample read sets for filling the gaps in the reconstruction to achieve a better coverage. The segmentation results in 2,859 markers with a total length of 3,143,627 bp, and we analyze 2,859 potential adjacencies observed in the extant genomes: 27 of these gaps have a length of 0, leaving 2,832 gaps to fill. All gaps of length 0 can be covered by reads, supporting the direct adjacency of the markers involved.

We can see in Figure 5.6 on page 85 that with the combined set of reads, we can fill almost all simple gaps by read sequences. In addition, we obtain a higher number

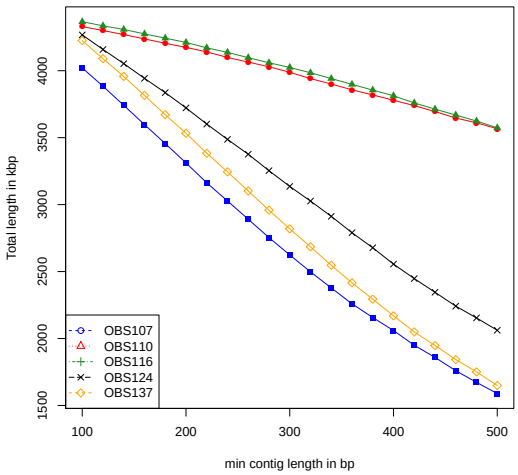


Figure 5.13: Total length of contigs mapped to *Yersinia pestis* CO92 greater than a minimum contig length.

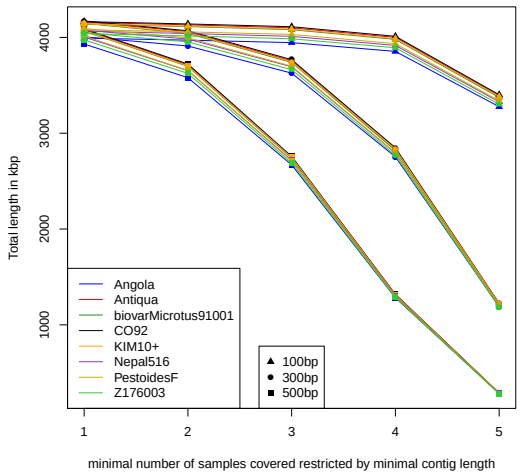


Figure 5.14: Comparison of the assembled contigs by mapping to different reference sequences. While most regions are covered by at least one sample, only a small part of them are covered by all five samples.

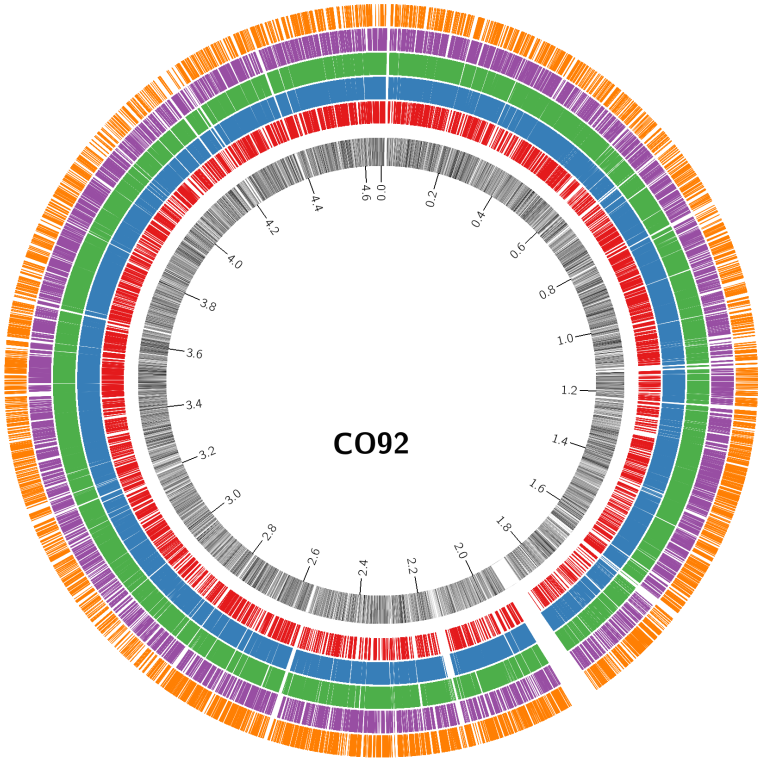


Figure 5.15: Mapping of contigs greater than 500 bp from all samples to *Yersinia pestis* CO92. The contigs from each sample are shown in a different color (from OBS117 in red to OBS137 in orange), indicating the different quality of the five samples. The reference is depicted in grey in the innermost ring, the shading indicates the number of samples covering a region in the reference genome.

of IS-annotated gaps that are filled in comparison to the London data set. For the IS-annotated gaps, 95 are reconstructed containing the IS, 22 contain IS annotations shorter than 400 bp. Hence we found the same number of potential ancestral IS as for the London strain.

We identified two conflicting components in this set of potential adjacencies (see Figure 5.16). Both of them align in terms of gap lengths and extant coordinates with the two components we observe with the de novo assembly for the London strain. In the first component, again only one conflicting adjacency is covered by reads. However, this is a different adjacency in comparison to both reconstructions for the London strain, while on the other hand we have no read support for the gap that is covered in the London data set. This could indicate a potential point of genome rearrangement (see discussion in the next section). In the second component, all involved adjacencies are covered by reads from the five samples. In order to obtain a set of high confidence

ancestral CARs, we removed all conflicting adjacencies in this component from the set of potential adjacencies. The coverage of all discarded adjacencies is shown in Figure 5.17.

Solving these conflicts results into 6 CARs for the ancient Marseille genome. Again, we used *BWA* [75] to align reads from all five samples to the assembly to assess the amount of uncovered regions in the reconstructed sequences. In total, only 54,672 bp in this mapping are not covered by any read and the length of the uncovered regions is rather short (see Figure 5.12 on page 93).

5.2.3 Comparison of both reconstructed ancient genomes

As the Marseille *Yersinia pestis* strain is assumed to be a direct descendant of the London Black Death strain [18], we aligned the obtained CARs of both de novo reconstructions to identify genome rearrangements [35, 73].

As shown in Figure 5.18, apart from one larger deletion and one larger insertion in the Marseille strain related to the removed gap sequence in the London strain and a small inversion of length 4,138 bp marked in black, the reconstructed CARs show no larger rearrangements between both genomes (grey links). The difference in conflicting adjacencies kept is a possible indication for a rearrangement that however cannot be explicitly identified at this point. It causes the split pattern observed between the CARs L3 and L1 in the London strain and M2 and M5 in the Marseille strain. Given that the available read data does not allow us to further order the resulting CARs into

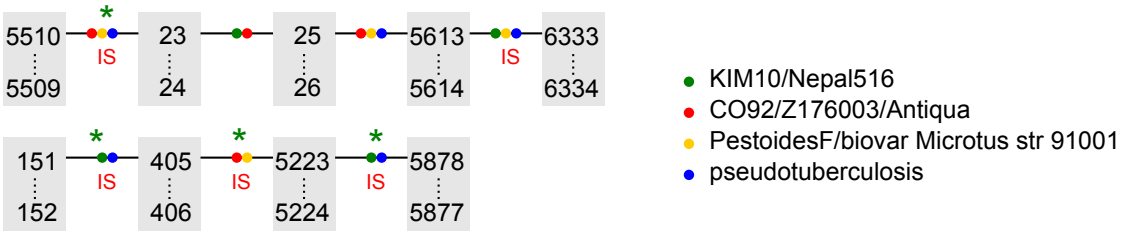
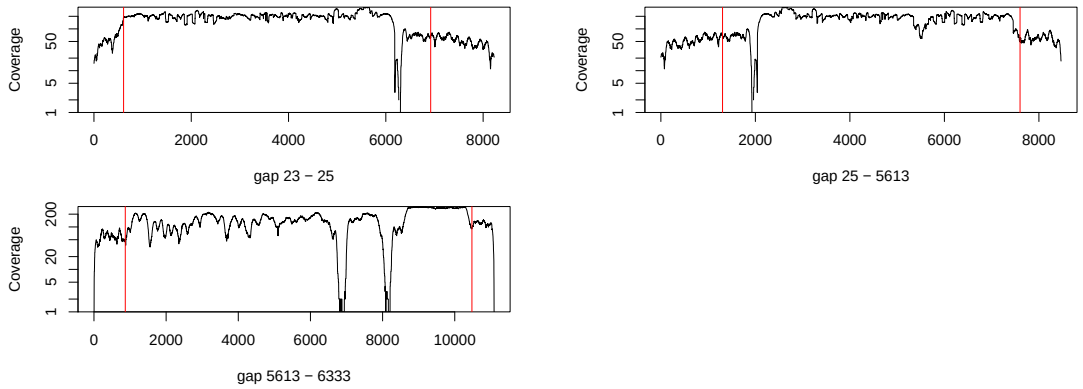


Figure 5.16: Conflicting components in the set of potential adjacencies for the Marseille data set. Markers are indicated by the grey boxes. Each marker x is represented by its extremities with $2x$ for the head and $2x - 1$ for the tail of the marker. The depicted IDs correspond to the real values in our data analysis. Adjacencies are depicted by connecting lines between two extremities. Gaps containing IS sequences are labeled accordingly. The color labels assigned to each adjacency indicate the extant occurrences and hence the conservation of the adjacency in the tree. All gaps that are fully covered by reads and do not contain breakpoints in the mappings are marked by green stars.

Component 1:



Component 2:

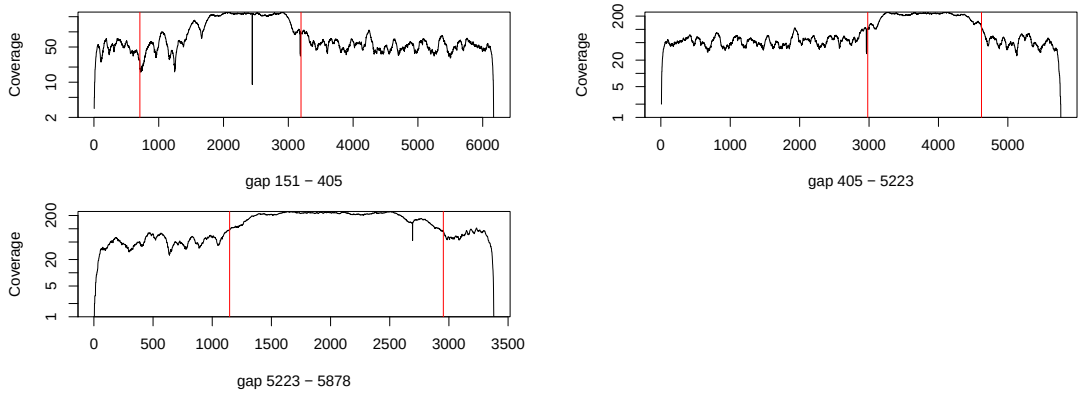


Figure 5.17: Read coverage for discarded adjacencies in conflicting components for the de novo reconstruction for the Marseille data set. The gap sequence is flanked by the marker, the gap borders are indicated in red.

a single scaffold, additional potential rearrangements could be assumed to be outside of the reconstructed CARs.

In contrast, Figure 5.18 depicts several inversions and translocations between both ancient sets of CARs and the extant *Yersinia pestis* CO92 strain (red and blue links respectively), confirming the high rate of rearrangements in the *Yersinia* phylogeny.

5.2.4 Discussion of local reconstruction

In this section, we have presented the analysis for two ancient *Yersinia pestis* strains isolated from the remains of victims of the second plague pandemic based on the local

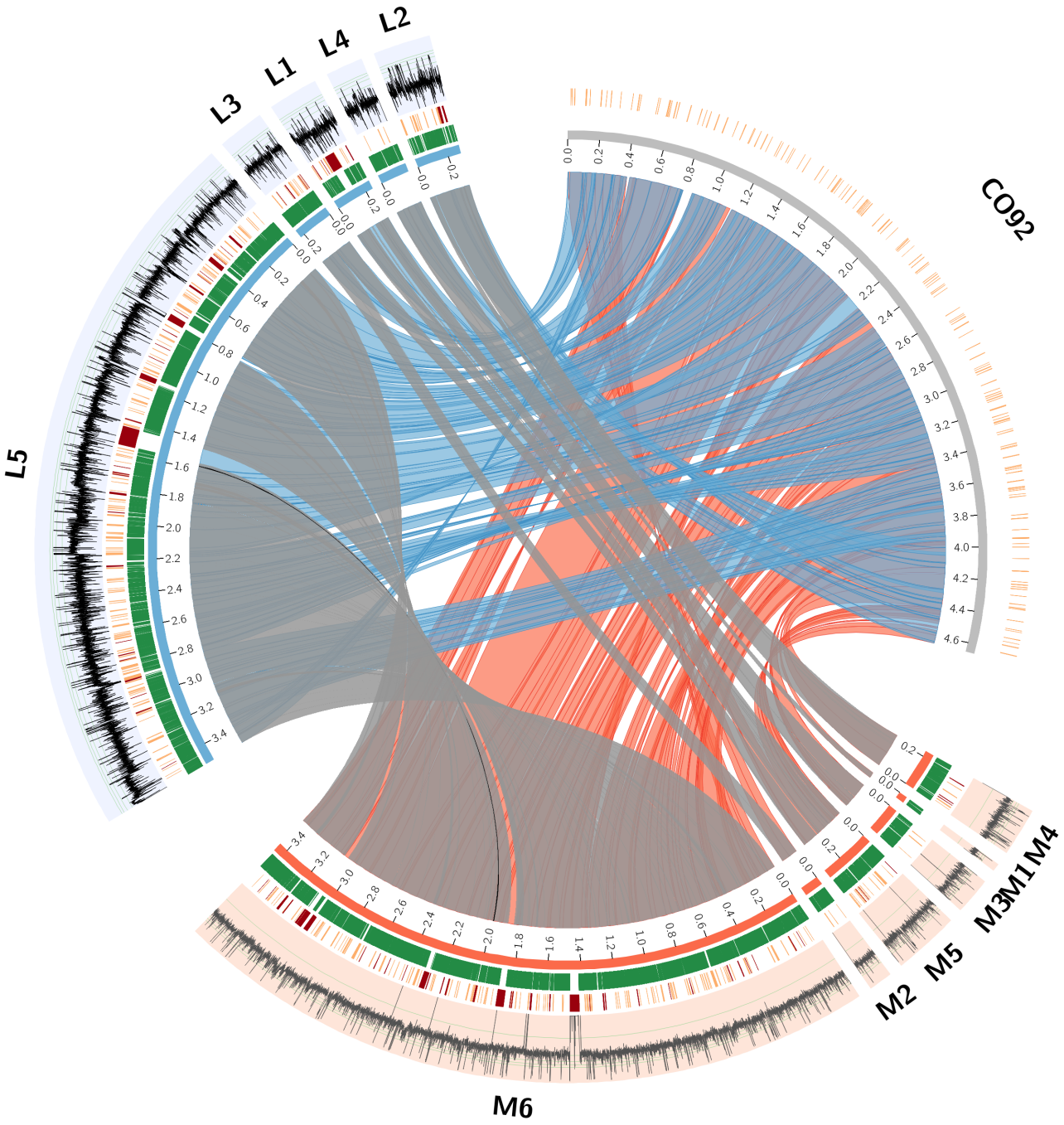


Figure 5.18: Comparison of the de novo assembly of the London strain (blue) and the Marseille strain (orange) with the reference *Yersinia pestis* CO92. The inner links connect corresponding CARs in the reconstructions and the reference. Note that there is only a small inversion marked in black among the grey links. The positions in both reconstructions covered by markers are indicated in green. All gaps that have IS annotations in the extant genomes are shown in orange. For CO92, all IS annotations are shown as well. In addition, gaps that are only partially filled or have very unconserved extant gap lengths are indicated in red. Finally, the outermost ring shows the average read coverage in windows of length 200 bp in log scale. The figure has been compiled with Circos [72].

reconstruction that combines comparative scaffolding using related extant genomes and direct aDNA sequencing data as described in Chapter 4.

The comparison of the reference-based and de novo assemblies for the London strain illustrates that relying on a shorter initial de novo contig assembly does not impact significantly the final result. In this case, we do not obtain any differences in terms of rearrangements between both assemblies, however using a reference during assembly involves the risk of missing rearrangements in the following analysis and hence being able to avoid a reference in the initial assembly is preferable. The results we obtain with the Marseille data set illustrate that if a good coverage of reads over the whole genome can be provided (as through multiple sequencing experiments for multiple samples), even a cautious initial contig assembly can be improved in such a way that most gaps are filled using unassembled aDNA reads. With both data sets, we obtain largely improved genome assemblies, with a reduced fragmentation (from thousands of contigs to a handful of CARs) and a very small fraction of the final assembly that is not supported by aDNA reads.

Applied to the same data set for the London strain, the method FPSAC [117] was able to obtain a single scaffold based on local parsimonious optimization. Comparing our resulting assembly to this single scaffold, we can identify two breakpoints between both assemblies, hence both methods do not entirely support the same scaffold structure for the London strain (see Subsection 5.5.1 for a detailed discussion). These disagreements should be seen as weak points in both solutions, as they are not reconstructed by different scaffolding objectives and would need to be confirmed more confidently by additional sequencing data.

We see a clear connection between conflicts in the set of potential adjacencies and the presence of IS elements in the corresponding gaps. Solving these conflicts based on aDNA read data provides a useful way to identify ancestral adjacencies in a conflicting component if the quality of the aDNA data is sufficient. The mapping of aDNA reads has shown to be mostly difficult at repetitive regions like Insertion Sequences, where the presence of the IS in the ancestral gap cannot be clearly detected by the aDNA sequencing data.

Interestingly, the improved assemblies of the London and Marseille strains show no explicit large genome rearrangements except for a small inversion. Even if potential genome rearrangement might not be observed due to the fragmentation of the assemblies into CARs, the synteny conservation between two strains separated by roughly 400 years of evolution is striking compared to the level of syntenic divergence with extant strains. This might be explained by the fact that both the London and Marseille

strains belong to a relatively localized, although long-lasting, pandemic [18]. Further, conflicting adjacencies in the Marseille data set were covered by aDNA reads, thus making it difficult to infer robust scaffolding adjacencies. This raises the question of the presence of several strains in the Marseille pandemic that might have differed by one or a few inversions.

Answering these questions with confidence would require additional targeted sequencing of a few regions of the genomes of the London and Marseille strains, or the sequencing of additional strains of the second plague pandemic, such as the recently sequenced *Yersinia pestis* genome from plague victims in Ellwangen [135] which is assumed to be an ancestor of the Marseille strains.

5.3 Global reconstruction of London ancestor with EWRA

While we presented a local reconstruction in the previous section, we use the aDNA data for the London Black Death agent in the global reconstruction approach described in Chapter 2 next. Again based on the phylogeny depicted in Figure 5.2 on page 79, we refer to this augmented ancestral node as the *London (L) node* in the following. Note that we do not consider the Marseille data at this point.

We assembled the reads with ABySS [133], an assembler for short read data based on a distributed de Bruijn graph implementation [31]. In comparison to several other short read assemblers including Minia [30], ABySS allows to output the graph after assembly. This graph then depicts assembled contigs and additional connections between contigs that could not be resolved during assembly. Given the short read length in the data, we set $k = 21$ as the k-mer length used to build the de Bruijn graph and option $-g$ to output the assembly graph needed as input for EWRA. The resulting assembly contains $3,018$ contigs with a length ≥ 500 bp. The contigs cover $3,104,032$ bp in total, while the N50 value for the assembly is $1,126$. The quality of this assembly hence lies somewhere in the middle between the reference-based and the Minia de novo assemblies described in Section 5.2, however the choice of assembly tools is limited by the requirement of the assembly graph in this analysis.

We used the segmentation process as described in [117] for the obtained contigs and all extant reference genomes to compute marker sequences, restricting the set of markers to be unique and universal. In total, we obtain $2,763$ marker families. On these markers, we also compute reconstructions using FPSAC [117] and the Fitch algorithm [54] as described in Chapter 2.

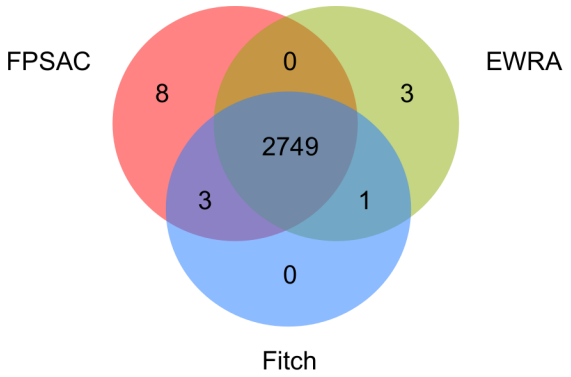


Figure 5.19: Venn diagram showing the number of marker adjacencies reconstructed at the London node by all three methods in comparison to the adjacencies only reconstructed by some of the approaches.

In order to use the assembly as input for EWRA, we additionally extract all adjacencies defined by the assembly graph as follows. We first locate all mappings of markers onto contigs (representing nodes in the assembly graph). Then we can easily find intra-contig adjacencies for all markers that are located on the same contig. For all markers that are mapped to the border of a respective contig, we can follow all edges in the de Bruijn graph connecting this contig to others in the graph, defining all adjacencies for potentially adjacent contigs. We obtain 124 adjacencies between markers within contigs, while we have 1,176 adjacencies defined by the edges between contigs in the assembly graph. As expected, the number of intra-contig adjacencies is low but still interesting, as they indicate either potential rearrangement breakpoints or assembly errors in terms of wrongly connected contiguous sequences. The set of inter-contig adjacencies is likely not complete, hence we do not restrict the scaffolding of the fragmented assembly to edges present in the assembly graph. It further contains 228 pairwise conflicting adjacencies that need to be resolved in the phylogenetic context.

We reconstruct 2,753 adjacencies at the London node with EWRA. All intra-contig adjacencies are reconstructed, confirming that there are no errors in the aDNA ABySS assembly. As all reconstructions are based on the same set of markers, we can directly compare the reconstructed adjacencies to the FPSAC and Fitch reconstructions as shown in Figure 5.19. We see a high agreement between the reconstructed sets with 2,749 adjacencies found by all three methods. The small differences however deserve to be discussed in more detail. We have one adjacency that is reconstructed by both global SCJ methods, but absent in the FPSAC reconstruction. The adjacency is present

in the *Yersinia pestis* strains Antiqua, Nepal516 and KIM10 but absent in the rest of the extant genomes. Hence global parsimony indicates the presence of this adjacency at the London node with a later loss along the branch to the strains CO92 and Z176003. In FPSAC however, for both extremities in this adjacency, conflicting adjacencies are reconstructed that are present in CO92 and Z176003 as it optimizes the weight-based objective in FPSAC to select a subset of adjacencies. Despite these, there are six other adjacencies that are only reconstructed in FPSAC, but have no support in a global reconstruction. In addition, none of these adjacencies are supported by the assembly graph.

We see three adjacencies with mixed signal in the extant genomes that are only reconstructed by EWRA, with conflicting alternatives reconstructed by FPSAC and the Fitch approach. For one of these adjacencies, we have support by the assembly graph, influencing the inclusion of this adjacency at the London node. More precisely, this adjacency would not be reconstructed without the positive signal in the assembly graph. For the other two, the edge lengths in the tree cause the scenario for these adjacencies to differ from the Fitch solution. This effect has already been observed with the mammalian data. Hence all these adjacencies only reconstructed by some of the methods are up for discussion when looking for a set of high confidence scaffolds, as these adjacencies do not have the global support in the tree when edge lengths are considered in the objective.

This analysis shows that the edge lengths considered in the EWRA method have an influence on the reconstruction despite being potentially able to reduce the fragmentation of the resulting scaffolds. We see a high agreement between the Fitch and the EWRA reconstruction when including adjacencies derived from an assembly graph of aDNA reads. These provide a way to confirm adjacencies with a mixed signal in the extant genomes. We also have a high agreement with the local FPSAC method, while also indicating adjacencies that should be classified as potentially weak in both reconstructions.

5.4 Global reconstruction of London ancestor with PhySca

Finally, we apply the adapted Sankoff-Rousseau method PhySca as introduced in Chapter 3. We will first give some results on the reconstruction considering only the London aDNA data. Besides the sequenced aDNA single-end reads for this strain, we build upon the reference-based assembly by Bos et al. [19] for London individual 8251, as it

provides the best assembly quality. Again, we will refer to this augmented ancestral node in the tree as the *London (L) node*.

The marker sequences for all extant genomes were computed as described in [117], restricting the set of markers to be unique and universal. For a total of 2,207 markers in all extant genomes we obtain 2,232 different extant adjacencies, thus showing a relatively low level of syntenic conflict compared to the number of markers, although it implies a highly dynamic rearrangement history over the short period of evolution [117].

As for the mammalian data set analyzed in Chapter 3, we considered as potentially ancestral any adjacency that appears in at least one extant genome. This does not restrict the set of potential adjacencies e. g. according to some specific parsimony model, but still does not allow the creation of adjacencies that are not observed in any extant genomes. Thus we are starting with a slightly higher number of potential adjacencies than in the analysis in Section 5.2. In contrast to the mammalian data set, where the phylogeny covers a larger evolutionary time, it is not necessary to reduce the complexity of connected components by applying a weight threshold x. We evaluate both approaches described in Subsection 3.2.6 on page 50 to weight adjacencies, starting with the weighting that is independent of the aDNA read data.

5.4.1 Ancestral reconstruction with Boltzmann weights

Fist, we computed Boltzmann weights for all internal nodes of the phylogeny, hence not considering the available aDNA data at this point. We sampled 500 solutions for different values of α each.

Recall that the parameter kT influences the weighting of the adjacencies: For $kT = 0.1$, the weights are based on sampling mostly optimal scenarios of presence and absence for the specific adjacency, while for $kT = 1$ also sub-optimal scenarios are likely to be sampled. The influence of this parameter on the distribution of adjacency weights is depicted in Figure 5.20. Again, we observe the specificity of weights with $kT = 0.1$, while the weights with $kT = 1$ are more balanced. In comparison to the mammalian data set however, we still see a larger amount of adjacencies with a weight close to 1 for $kT = 1$, hinting at the different complexity of this data set in comparison to the mammalian data.

Figure 5.21 displays the number of reconstructed scaffolds for some selected nodes in the phylogeny (see Figure 5.2 on page 79). For $\alpha = 0$, the number of scaffolds minimizing the SCJ distance over the whole tree varies at all considered internal nodes.

For higher values of α however, the number of scaffolds becomes robust and is generally decreasing with the increase of α, as the inclusion of adjacency weights favors the presence of adjacencies at internal nodes in combination with the overall tree costs. Interestingly, for $kT = 0.1$, the solutions are robust over all $0 < \alpha < 1$, indicating that the importance of both parts of the objective function is variable over the resulting solutions. As the weights for $kT = 1$ are in general less precise, the solutions are only robust for specific values of α.

While the number of scaffolds decreases, the total SCJ distance in the tree increases with increasing values of α as seen in Figure 5.22. Regarding the London node, the comparative approach is able to reduce the number of scaffolds to a maximum of 15 and a minimum of 1 when only the adjacency weights are considered, confirming the result in [117].

5.4.2 Ancestral reconstruction with aDNA weights

For the London node, adjacency weights can be based on the given aDNA reads for each given potential ancestral adjacency. In order to solely observe the effect of the aDNA weights in the global reconstruction, we assign a weight of 0 for all adjacencies at other nodes in the tree. Moreover, this weighting scheme addresses the issue of

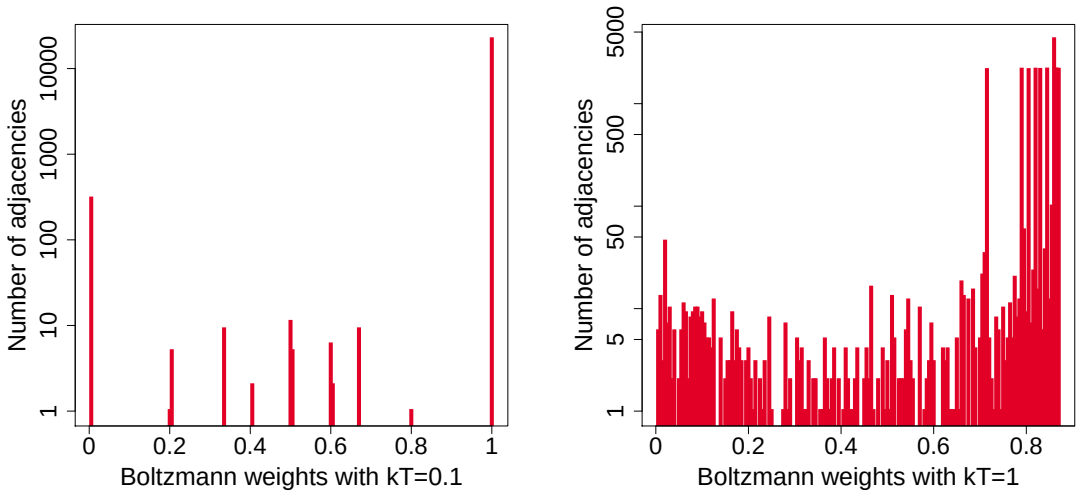


Figure 5.20: Distribution of the Boltzmann weights for the London *Yersinia pestis* data set for all potential adjacencies with $kT = 0.1$ and $kT = 1$.

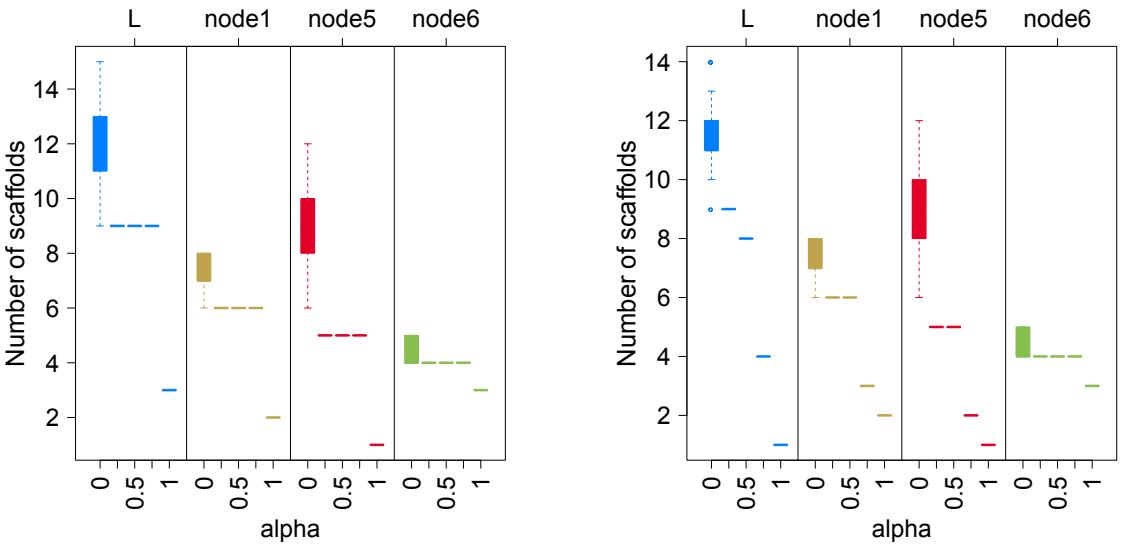


Figure 5.21: *Yersinia* data set: Reconstructed number of scaffolds with Boltzmann weights at $kT = 0.1$ (left) and $kT = 1$ (right).

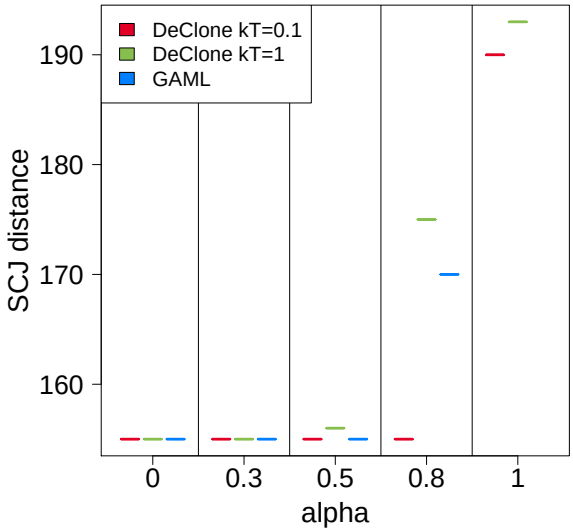


Figure 5.22: Total SCJ Distance in the tree for Boltzmann weights computed with DeClone and aDNA weights according to the GAML model.

potential adjacencies at the London node with a lower weight due to the difficulty of sequencing ancient DNA.

As expected, the aDNA weights are more skewed to the higher end of the scale than the Boltzmann weights as displayed in Figure 5.23. Only one adjacency has a

weight close to 0, indicating that nearly all gaps have at least some reads mapping to the proposed template gap sequence. As already observed in the gap filling analysis in Section 5.2, a high amount of ancestral gaps can actually be covered completely with read sequences, resulting in weights close to 1 and corresponding to the peak observed in Figure 5.23.

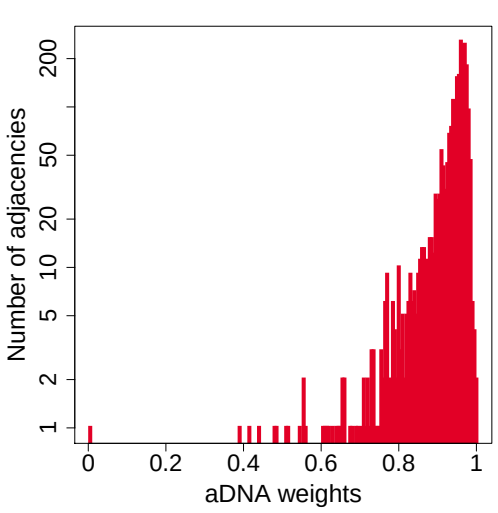


Figure 5.23: Distribution of the aDNA weights for all potential adjacencies at the London node.

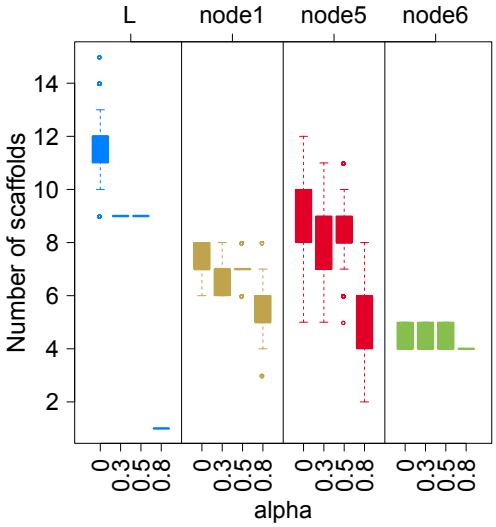


Figure 5.24: Reconstructed number of scaffolds in the *Yersinia* data set with aDNA weights at the London node and 0 otherwise, for four ancestral nodes.

Again we sampled 500 solutions for this data set under different values of α. As shown in Figure 5.24, for selected internal nodes of the phylogeny, the pure SCJ solutions at $\alpha = 0$ result in the highest fragmentation, while the number of CARs decreases as we increase the importance of the adjacency weights in the objective of our method. For the London node, when including the aDNA weights, the fragmentation is decreasing while the reconstructions for each $\alpha > 0$ are robust. At the other nodes, the applied sequencing weights also reduce the fragmentation except for node 6 which is located in the *pseudotuberculosis* subtree and hence more distant to the London node. This shows that the aDNA weights not only influence the reconstructed adjacencies at the London node, but also other nodes of the tree.

5.5 Global reconstruction of London and Marseille strains

The method described in Chapter 3 allows the joint global scaffolding of multiple ancient assemblies. We will use the previously separately analyzed ancient *Yersinia pestis* strains from the Marseille [18] and London outbreak of the bubonic plague [19] and build a combined set of markers using assembled contigs from both strains. We can then compute the adjacency weights for all potential adjacencies based on both aDNA read sets and apply the Sankoff-Rousseau framework to obtain a global reconstruction using both sources of information. The Marseille strain represents an extinct leaf in the phylogeny. Eventually, we will compare reconstructed genome sequences with the sequences obtained by the AGapEs pipeline [78] as described in Section 5.2 and FPSAC [117].

Defining marker families

Given two sets of ancient contigs, the first step is to define a common set of markers supported by each set of contigs and the set of extant genomes. Ideally, we search for non-overlapping markers that have a unique occurrence in each of the extant genomes and match at least one contig in both ancient data sets. For this task, we adapt the iterative segmentation approach described in [117]. As it is based on pairwise alignments between contigs and the extant reference genomes, we can extend the segmentation by also aligning the sets of contigs against each other, then filtering for families that are unique and universal in the extant genomes and are additionally supported by both ancient data sets. This approach is obviously not very efficient as a lot of pairwise alignments have to be computed. It is further hard to extend as soon as additional ancient assemblies are included or the quality of the ancient assemblies is differing substantially. A more general framework based on a pan-genome data structure would be a point of research for the future as discussed in Chapter 6.

As input, we consider both de novo assemblies of the London and Marseille strains and the extant reference genomes as described in Section 5.2, as well as the phylogeny depicted in Figure 5.2 on page 79. The segmentation results in 2750 marker families with a minimal length of 100 bp, covering 1,786,815 bp in total. Hence with the requirement that each family contains a contig of each ancient data set, we lose around 400.000 bp in comparison to the marker set for the de novo assembly of the London strain alone.

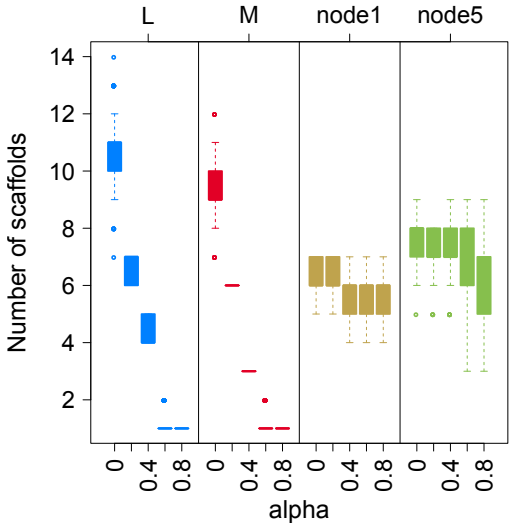


Figure 5.25: Number of reconstructed scaffolds for different values of α on the joint marker set for the London and Marseille ancient genomes.

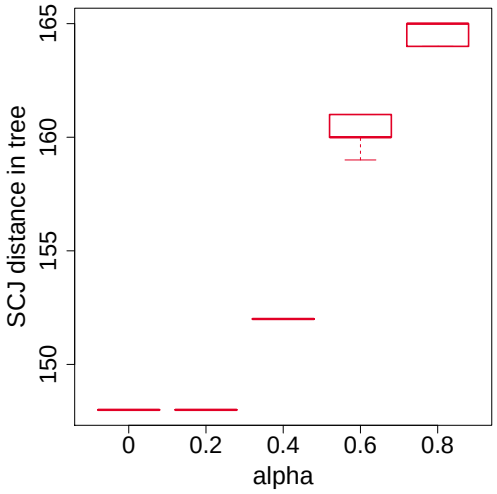


Figure 5.26: SCJ distance in the tree for different values of α.

Reconstruction

We sampled 500 solutions for each α between 0 and 1 in steps of 0.2. As the complexity of the data allows us to omit a weight threshold, we identify 2774 potential adjacencies at each ancestral node. Among them, we have 51 conflicting adjacencies.

Figure 5.25 shows the number of reconstructed scaffolds for the London and Marseille nodes as well as one ancestor (node 5) and one descendant (node 1) in the phylogeny. We can see a gradual decrease in the number of scaffolds for both nodes with aDNA weights, but in contrast to the Marseille node, we do not get robust results for the London node over different values of α as seen in the previous section. For $\alpha = 0$, the range of number of scaffolds in the sampled solutions is lower for the Marseille ancestor, and the solutions are more robust for higher values of α. This is not surprising, as the set of adjacencies reconstructed for this leaf is not influenced in the bottom-up traversal of the tree. For values of α close to 1, we obtain a single scaffold for both ancient nodes, while we still obtain several scaffolds at the other unweighted nodes.

As seen before, with the decrease in the number of scaffolds, we see an increase in the total SCJ tree distance. In comparison to the previous analysis however, while the distance is constant over all samples for smaller α, we sample solutions with slightly

differing tree distances for higher values of α. It shows that the weighting of two nodes in the tree based on aDNA data results in co-optimal solutions that are balancing differing tree cost. We did not observe this when we weighted only one internal node based on the aDNA as seen in Figure 5.22 on page 105.

5.5.1 Comparison to FPSAC and AGapEs

In this section, we compare the reconstructed nucleotide sequences for the London and the Marseille strain for different approaches. While FPSAC [117] uses the aDNA sequencing data only in the form of assembled contigs and afterwards applies parsimonious optimization for the scaffolding, with AGapEs we described a combinatorial method that substitutes this optimization steps (solving conflicts, filling gaps) by using the unassembled aDNA reads. We compare these two results with the PhySca reconstructions presented in the previous section, where the global tree distance and the aDNA read information are combined in a global reconstruction. In all PhySca reconstructions, we also filled the gaps in the already consistent marker orders with AGapEs in order to compare complete nucleotide sequences.

We identify two rearrangements between the proposed scaffolds that have to be scrutinized to find the most reliable ancestral genome for the London ancestor. We illustrate the alignments of the resulting scaffolds for the London strain as synteny plots computed with r2cat [67] in Figure 5.27. Comparing the reconstructions of FPSAC and AGapEs, we see two rearrangements that are marked in red and green respectively. We analyze these differences by comparing them with the PhySca results over different values of α. We also marked two noticeable unaligned regions observed in the alignment, one due to the removed unconserved gap sequence in the AGapEs reconstruction and the other due to the reconstructed IS sequence that is not seen in the parsimonious Fitch gap sequence reconstructed by FPSAC.

The rearrangement marked in red shows a transposition between the FPSAC and the AGapEs reconstruction. In the PhySca reconstruction, we find the AGapEs variant for smaller values of α, e. g. $\alpha \leq 0.3$, and reconstruct the FPSAC variant accordingly for higher values of α, e. g. as marked in red for $\alpha = 0.5$. At a first glance, this seems counter-intuitive, as higher values of α lay the importance on the adjacency weights based on read data and this data is likewise used to solve the conflicts in AGapEs. However, this can be explained by a closer look at the underlying conflicting component: the red rearrangement is caused by the conflicting component identified in the AGapEs analysis consisting of three conflicting adjacencies (see Figure 5.8 on page 88).

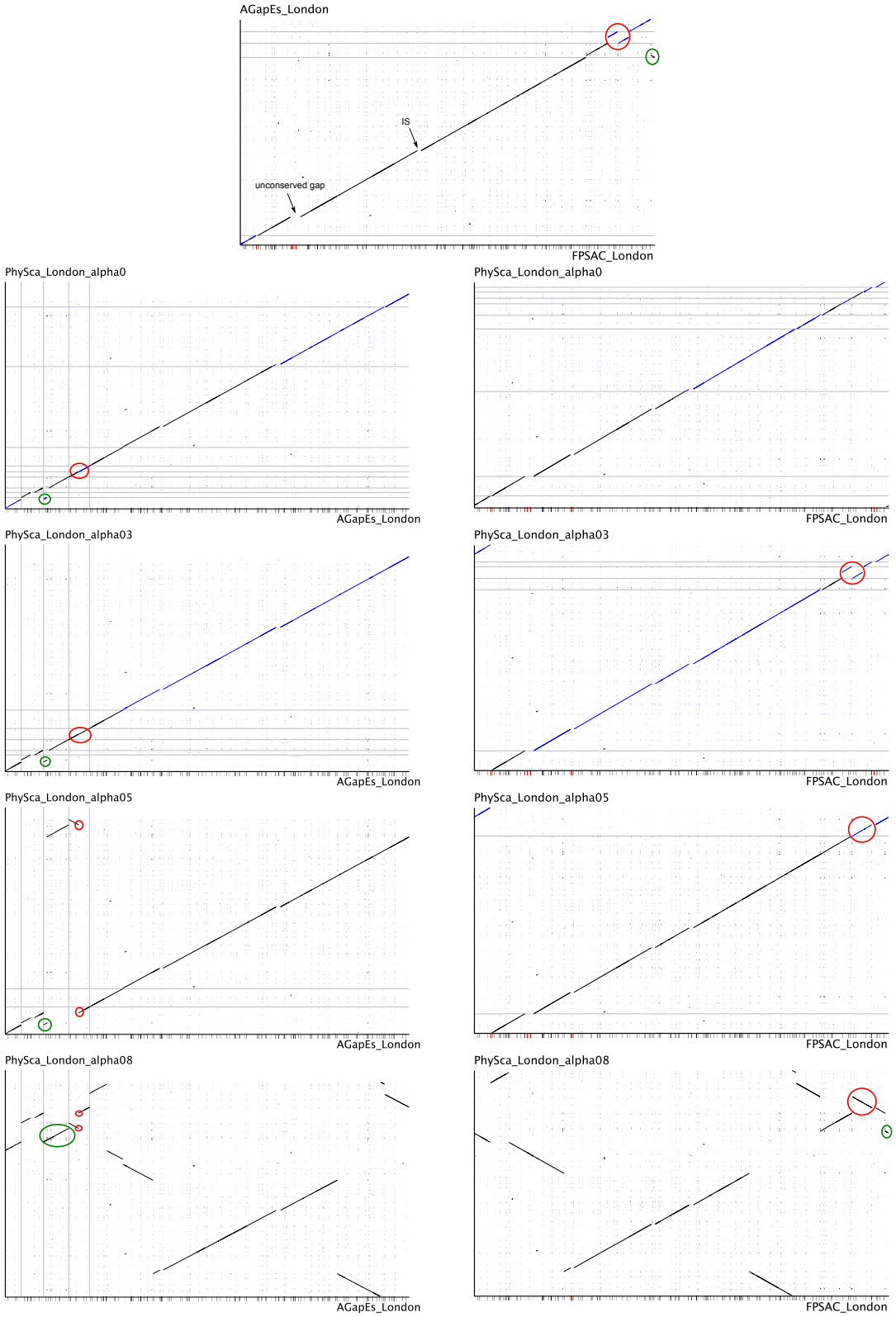


Figure 5.27: Visualization of alignments between reconstructions for the London ancestor by FPSAC and AGapEs (top), PhySca and AGapEs (left column), and PhySca and FPSAC (right column). Synteny plots computed with r2cat [67].

Based on the read coverage for these adjacencies, in AGapEs we include one of these adjacencies and discard the other two. In fact, this kept adjacency also has a higher adjacency weight in PhySca as expected. As all of these adjacencies are only present in a fraction of the extant genomes, the PhySca objective then keeps the higher weight adjacency only in combination with the global SCJ costs. For higher α however, it becomes optimal to keep both flanking adjacencies with a lower weight and to discard only a single adjacency, which also corresponds to the maximum weight objective used in FPSAC to solve conflicts.

The small inversion marked in green shows the opposite trend: For lower values of α, even for $\alpha = 0$ which represents the most fragmented solution in PhySca, we see a difference in comparison to the AGapEs reconstruction, hence confirming the FPSAC variant at this point. Only with a high emphasis on the adjacency weights with $\alpha = 0.8$, the marker order as seen in AGapEs is reconstructed. It indicates an adjacency with a high weight that is however also causing high evolutionary costs in the tree.

For smaller values of α, PhySca is not reconstructing any differences that are not seen in either AGapEs or FPSAC. Especially the longer scaffold that can be seen for $\alpha = 0.3$ and 0.5 is consistent over all reconstruction methods, allowing to assume a high confidence in this part of the reconstructed genome sequence. However, for higher values of α, the alignments also visualize the drawback of only focusing on adjacency weights in the optimization, as we see a lot of differences to the other reconstructions. Even though with $\alpha = 0.8$ we still consider the evolutionary tree costs in the objective, the weights are already superimposing the tree costs (also, there is no difference to the solution with $\alpha = 1$). Hence for high α, it indicates that the adjacency weights based on the GAML model are not complying to the parsimony assumptions for some conflicting adjacencies. This can be due to missing data in the aDNA reads, which is not accounted for in the weighting of the adjacency. Also note that the AGapEs analysis is mainly focused on uncovered regions and breakpoints in the mappings of the aDNA data. While these regions surely influence the weighting in the GAML model, it is however less emphasized compared to the weights of adjacencies that are fully covered by reads. This shows that we need to combine these weights with the evolutionary tree model in order to receive reliable results in the reconstruction.

In the PhySca setting, the global reconstruction of the Marseille strain being an extinct leaf is only directly influenced by the parent node, i. e. the London strain, but not by any other extant leaves in the tree. We do not observe any rearrangements between both ancient strains in the PhySca reconstructions, confirming the high conservation

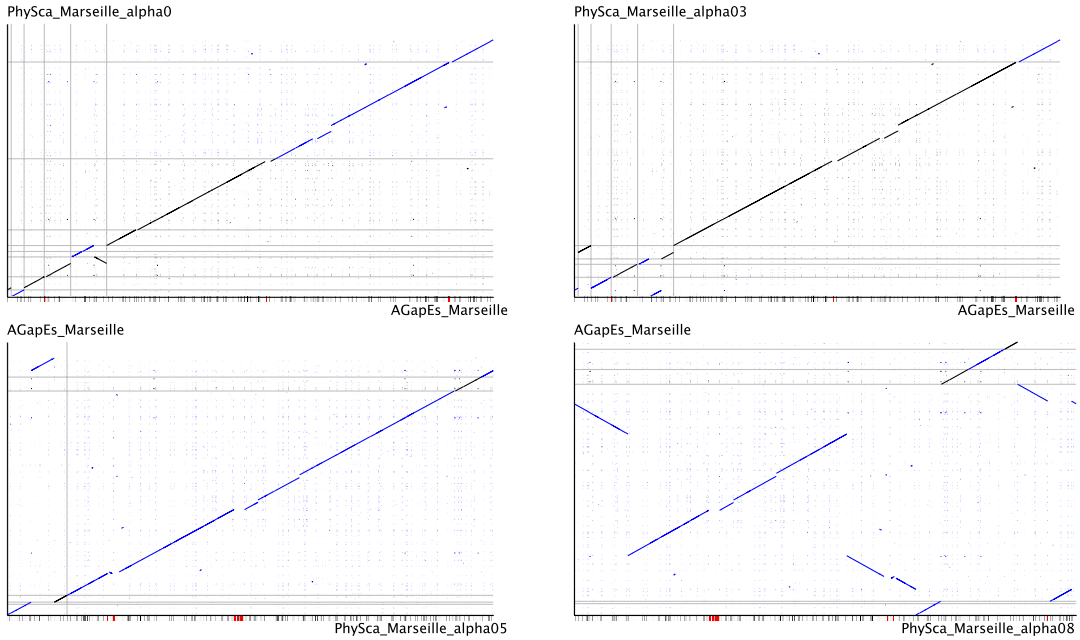


Figure 5.28: Visualization of alignments between reconstructions for the Marseille data set by AGapEs and PhySca. The less fragmented solution is taken as the reference. Synteny plots computed with r2cat [67].

between both strains as seen in the AGapEs analysis. Comparing the reconstructions for the different methods in Figure 5.28, we see the same differences as observed for the London strain, due to the different emphasis on missing data and uncovered regions. Again, we see a few differences between the AGapEs reconstruction and the PhySca reconstruction for small values of α. Interestingly, for $\alpha = 0.5$, the difference observed in the synteny plot is caused by the circularity of the chromosome and no other larger rearrangement can be observed. For $\alpha = 0.8$, the differences are very similar to the differences we already pointed out for the London strain, indicating that they cannot be caused by the aDNA data alone but are rather imputable to the methods themselves, given that the Marseille reconstruction in PhySca is highly influenced by the reconstruction of the London strain.

5.5.2 Discussion of compared reconstructions

A reconstructed ancestral genome is a complex result that cannot be easily validated. Local and global methods with a different emphasis on the inclusion of the aDNA data provide a useful way to identify points in the reconstructions that are questionable. For the reconstruction of the London ancestor in the *Yersinia pestis* phylogeny, we point

out two weak adjacencies, that can also provide hints at weak points of the different methods presented. Especially the AGapEs method depends on the quality of the aDNA sequencing data, while the optimization framework in FPSAC does not account for potential convergent evolution. Parameter α in the PhySca method provides a useful way to explore different reconstructions with varying emphasis, as can be seen in the comparison with FPSAC and AGapEs, where the solutions for different values of α can agree with one method or the other.

In comparison to the PhySca method in Chapter 3, AGapEs is based on a purely combinatorial approach, while adjacency weights in PhySca are computed using the aDNA data in a probabilistic framework. In comparison, the AGapEs approach relies on uncovered regions and mapping breakpoints alone, hence scaffolding a fragmented assembly mainly concentrated on these regions. On the other hand, the aDNA weights in PhySca do reflect such regions in the aDNA data by lower weights respectively, but include them in the weighting based on the coverage observed for the whole gap. The comparison between both approaches here illustrates these differences.

On a positive note, the number of weak points in relation to the large part of the ancient genomes that can be reconstructed confidently is small and indicates that the assumptions underlying the different methods are well grounded. Again, additional sequencing data for these regions of the reconstructions can help to clarify the most likely variant, while especially the global reconstruction methods can show its true value if even more ancient DNA data sets become available (e. g. very recent sequencing data for additional ancient *Yersinia pestis* genomes [49, 135]) and can be integrated in the analysis.

Chapter 6

Conclusion and Perspectives

In this thesis, we investigated the joint question of ancestral genome reconstruction and scaffolding of fragmented aDNA assemblies. Based on a given phylogeny and extant genomes as a common basis for both problems, we explored integrative phylogenetic methods to solve the Small Parsimony Problem with different approaches to include aDNA data in the optimization, and also presented a local method to close gaps in proposed marker orders based on aDNA as the last step of scaffolding.

At first in Chapter 2, we generalized the result of [46] regarding the SCJ Small Parsimony Problem towards multifurcating trees with edge lengths, while we showed that consistency of the solutions can still be guaranteed and we can expect to obtain a unique optimal solution under non-trivial edge lengths. Building upon this result, we presented EWRA, an integrative approach including one aDNA sequencing data set in the form of a contig assembly graph. The global reconstruction of all ancestors in the tree by minimizing a distance based on the parsimony principle also provides a scaffolding of the fragmented ancient assembly in the same time.

In Chapter 3, we extended the previous approach by allowing the inclusion of local information at different nodes of the phylogeny, e. g. provided through several aDNA sequencing data sets. We defined the SCJ Small Parsimony Problem with locally weighted marker adjacencies. The optimization problem is described through an objective that – based on a presence/absence representation of adjacencies – combines the global evolutionary cost and local weights for absent adjacencies at specific nodes in the tree. Both terms are integrated through a convex combination factor that allows to explore the influence of both the global tree evolution and the local adjacency weights on the solutions. We presented a fixed-parameter tractable algorithm based on an adaption of the Sankoff-Rousseau method to find a solution under this objective,

enabling also to extend the dynamic-programming framework to sample co-optimal solutions to explore the solution space. We investigated two ways of weighting adjacencies locally in the tree: Through independent sampling of adjacencies scenarios under the Boltzmann distribution and through the mapping of aDNA reads to potential marker adjacencies.

While these two methods are based on global ancestral reconstruction in a parsimony framework, in Chapter 4 we mainly concentrated on available aDNA read data for specific genomes of interest in the tree. We described the problem to fill gaps between adjacent markers with overlapping sequences of aDNA reads, minimizing the distance to a template sequence. We illustrated a straight-forward algorithm based on Dijkstra's shortest path algorithm in a graph defined by overlapping read mappings. Based on this approach, we introduced the pipeline AGapEs to benefit from the estimation of ancestral gap sequences in order to clear conflicts and analyze specific features for a set of potential adjacencies for an ancient genome.

In Chapter 5, we put the theory into practice and presented a detailed analysis of extant and ancient *Yersinia pestis* strains. Applying all methods presented in this thesis to two ancient strains in a global phylogenetic context and comparing the resulting reconstructions facilitated us to explore the strengths and weaknesses of the different approaches. The diverse array of strategies let us identify confidently reconstructed parts of the ancient *Yersinia pestis* genomes as well as weak points in the reconstructions.

With the methods presented in this thesis, we illustrated the connection between ancestral genome reconstruction and scaffolding of aDNA assemblies as stated jointly in Question 3 (page 19) from both angles. The first two methods are based on global ancestral reconstruction principles and include the aDNA data as additional input to the optimization problem. In PhySca, our objective then allows to vary the importance of either the global evolutionary cost in the tree or the integrated local aDNA data. In other words, both methods start from a comparative ancestral reconstruction and we extended them to integrate the scaffolding of the fragmented aDNA assembly. With the local gap filling method, we started with a marker order based on a fragmented assembly of the aDNA data and included the comparison of related extant genomes in the phylogenetic context to find ancestral gap sequences. Such template sequences constructed parsimoniously from extant gap sequences allow to fill the gaps and analyze conflicting adjacencies as well as potential breakpoints in the same time. Both directions pave the way towards a fully integrated phylogenetic scaffolding method

that combines an evolutionary model and sequencing data for selected extant and ancestral genomes.

Perspectives The research presented in this thesis gives rise to several extensions and generalizations that could improve and enhance solutions to the integrated phylogenetic assembly problem.

A first exciting perspective is the availability of new sequencing data for ancient genomes that we can include in the reconstruction analysis in a phylogenetic context. Especially the global methods presented in this thesis can benefit from additional sequenced ancient strains in the same phylogeny, as e. g. in PhySca, we can provide more local weights based on aDNA data to guide the reconstructions. Also the local reconstructions by AGapEs can be evaluated more reliably, as additional reconstructed genomes can confirm the rearrangement history described in the tree. To this end, additional data can entail better evaluation methods to provide an improved notion of confidence in the reconstruction results, as the quality of methods estimated by simulation experiments is often biased.

Especially for the *Yersinia pestis* phylogeny, given the special interest of several research fields to unravel the history of the ancient plague pandemics, additional sequencing data sets – some even with increased sequencing quality - were already published very recently [49, 135] and hence provide an interesting perspective to extend the analysis on this specific bacteria family presented here.

With the inclusion of more ancient data, a crucial first step is the definition of marker families supported by extant genomes and several ancient sets of contigs in a combined problem statement of scaffolding and reconstruction. The methods presented here are relying on unique and universal markers so far, hence the amount of markers that can be defined with the support of all sets of contigs will decrease rapidly with the number of assemblies considered. Especially different sequencing quality and conservation of the ancient material can make it harder to define unique and universal marker families with a good coverage of the expected genome size and efficient methods to compute marker families are needed.

Naturally, this problem is closely related to the analysis of pan-genomes, where several data structures have been developed to store and analyze the core and dispensable genomes of several related strains or species. If these data structures allow an efficient extraction of the core genome, we can define markers on this core as input for the rearrangement analysis. Especially data structures that do not require assembled genomes as input, e. g. the BFT [64] which is based on the decomposition of reads

into k-mers, could even allow to avoid a first assembly of the aDNA reads and hence provide a way to find markers when several ancient read sets are available, however the efficiency of assembly and marker definition in one simultaneous step has to be investigated. Ideally, if a pan-genome representation is able to also take the relations between strains given in the phylogenetic tree into account, markers could be defined much more generally during the extraction of the core genome and we could avoid fragmentation that is only caused by missing data in the ancient reads.

Besides generally improving the step of marker definition, the methods presented in this thesis assume several restrictions to the underlying modeling of the data that can be generalized in order to extend the coverage of genomes by markers as input to the reconstruction methods. Deviating from the notion of unique and universal markers in the optimization would allow to be less strict and to include repeated regions of the genomes in the context of duplicated markers [122], however increasing the complexity of the problems defined on this genome model. Gene trees provide an opportunity to integrate duplicated markers in the reconstruction of genome structure and it is interesting to extend the problems discussed here in this direction. The models could also be extended towards other instances than adjacencies, i. e. gene clusters as groups of more than two markers, following the reconstruction framework introduced in [138]. However the notions of conflicts and consistency are not as easily defined and identified as for simple adjacencies [105,150], again increasing the complexity that has to be handled to find solutions.

Besides the genome model, we used the parsimony principle to model evolution throughout this thesis. An alternative to explore are probabilistic approaches as for example in [65,81] defining a maximum-likelihood based objective. Such probabilistic solutions are interesting to compare to the parsimony-based reconstructions presented here.

All in all, the perspectives for integrated phylogenetic scaffolding and reconstruction are driven by the availability of ancient data sets that are exciting to explore from a practical point of view. From a theoretical and methodological point of view, developments of the underlying models for genomes and evolution in order to integrate aDNA data open several research avenues that are to be addressed further in the future.

Bibliography

[1] M. Achtman. How old are bacterial pathogens? *Proceedings of the Royal Society of London B: Biological Sciences*, 283(1836), 2016.

[2] M. Achtman, G. Morelli, P. Zhu, T. Wirth, I. Diehl, B. Kusecek, A. J. Vogler, D. M. Wagner, C. J. Allender, *et al.* Microevolution and history of the plague bacillus, Yersinia pestis. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17 837–17 842, 2004.

[3] Z. Adam and D. Sankoff. The ABCs of MGR with DCJ. *Evolutionary Bioinformatics*, 4, 2008.

[4] S. Aganezov, N. Sitdykova, and M. A. Alekseyev. Scaffold assembly based on genome rearrangement analysis. *Computational Biology and Chemistry*, 57:46–53, 2015.

[5] M. Alekseyev and P. A. Pevzner. Breakpoint graphs and ancestral genome reconstructions. *Genome Research*, gr–082 784, 2009.

[6] Y. Anselmetti, V. Berry, C. Chauve, A. Chateau, E. Tannier, and S. Bérard. Ancestral gene synteny reconstruction improves extant species scaffolding. *BMC Genomics*, 16 Suppl 10:S11, 2015.

[7] Y. Anselmetti, N. Luhmann, S. Bérard, E. Tannier, and C. Chauve. Comparative Methods for Reconstructing Ancient Genome Organization. *Methods in Molecular Biology*, 2016, accepted.

[8] P. Avdeyev, S. Jiang, S. Aganezov, F. Hu, and M. A. Alekseyev. Reconstruction of ancestral genomes in presence of gene gain and loss. Tech. Rep. 3, 2016.

[9] V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.

[10] S. Bérard, C. Gallien, B. Boussau, G. J. Szöllősi, V. Daubin, and E. Tannier. Evolution of gene neighborhoods within reconciled phylogenies. *Bioinformatics*, 28(18):i382–i388, 2012.

[11] A. Bergeron, J. Mixtacki, and J. Stoye. A unifying view of genome rearrangements. In *International Workshop on Algorithms in Bioinformatics*, 163–173. Springer, 2006.

[12] C. Berthelot, F. Brunet, D. Chalopin, A. Juanchich, M. Bernard, B. Noël, P. Bento, C. Da Silva, K. Labadie, *et al.* The rainbow trout genome provides novel insights into evolution after whole-genome duplication in vertebrates. *Nature Communications*, 5, 2014.

[13] D. Bertrand, Y. Gagnon, M. Blanchette, and N. El-Mabrouk. Reconstruction of ancestral genome subject to whole genome duplication, speciation, rearrangement and loss. In *International Workshop on Algorithms in Bioinformatics*, 78–89. Springer, 2010.

[14] P. Biller, P. Feijão, and J. Meidanis. Rearrangement-Based Phylogeny Using the Single-Cut-or-Join Operation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(1):122–134, 2013.

[15] D. Birnbaum, F. Coulier, M.-J. Pébusque, and P. Pontarotti. "Paleogenomics": Looking in the past to the future. *Journal of Experimental Zoology*, 288(1):21–22, 2000.

[16] M. Blanchette. Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Research*, 14(12):2412–2423, 2004.

[17] K. I. Bos, K. M. Harkins, A. Herbig, M. Coscolla, N. Weber, I. Comas, S. A. Forrest, J. M. Bryant, S. R. Harris, *et al.* Pre-columbian mycobacterial genomes reveal seals as a source of New World human tuberculosis. *Nature*, 514(7523):494–497, 2014.

[18] K. I. Bos, A. Herbig, J. Sahl, N. Waglechner, M. Fourment, S. A. Forrest, J. Klunk, V. J. Schuenemann, D. Poinar, *et al.* Eighteenth century Yersinia pestis genomes reveal the long-term persistence of an historical plague focus. *eLife*, e12994, 2016.

[19] K. I. Bos, V. J. Schuenemann, G. B. Golding, H. A. Burbano, N. Waglechner, B. K. Coombes, J. B. McPhee, S. N. DeWitte, M. Meyer, *et al.* A draft genome of Yersinia pestis from victims of the Black Death. *Nature*, 478(7370):506–510, 2011.

[20] E. Bosi, B. Donati, *et al.* MeDuSa: a multi-draft based scaffolder. *Bioinformatics*, 31(15):2443–2451, 2015.

[21] G. Bourque and P. A. Pevzner. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Research*, 12(1):26–36, 2002.

[22] V. Boža, B. Brejová, and T. Vinař. GAML: genome assembly by maximum likelihood. *Algorithms for Molecular Biology*, 10(1), 2015.

[23] J. H. Camin and R. R. Sokal. A method for deducing branching sequences in phylogeny. *Evolution*, 311–326, 1965.

[24] L. L. Cavalli-Sforza and A. W. Edwards. Phylogenetic analysis. Models and estimation procedures. *American Journal of Human Genetics*, 19(3 Pt 1):233, 1967.

[25] P. S. Chain, E. Carniel, F. W. Larimer, J. Lamerdin, P. Stoutland, W. Regala, A. Georgescu, L. Vergez, M. Land, *et al.* Insights into the evolution of Yersinia pestis through whole-genome comparison with Yersinia pseudotuberculosis. *Proceedings of the National Academy of Sciences of the United States of America*, 101(38):13 826–13 831, 2004.

[26] P. S. G. Chain, P. Hu, S. A. Malfatti, L. Radnedge, F. Larimer, L. M. Vergez, P. Worsham, M. C. Chu, and G. L. Andersen. Complete Genome Sequence of Yersinia pestis Strains Antiqua and Nepal516: Evidence of Gene Reduction in an Emerging Pathogen. *Journal of Bacteriology*, 188(12):4453–4463, 2006.

[27] C. Chauve, H. Gavranovic, A. Ouangraoua, and E. Tannier. Yeast Ancestral Genome Reconstructions: The Possibilities of Computational Methods II. *Journal of Computational Biology*, 17(9):1097–1112, 2010.

[28] C. Chauve, Y. Ponty, and J. P. Zanetti. Evolution of genes neighborhood within reconciled phylogenies: an ensemble approach. *BMC Bioinformatics*, 16(Suppl 19):S6, 2015.

[29] C. Chauve and E. Tannier. A Methodological Framework for the Reconstruction of Contiguous Regions of Ancestral Genomes and Its Application to Mammalian Genomes. *PLoS Computational Biology*, 4(11):e1000 234, 2008.

[30] R. Chikhi and G. Rizk. Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms for Molecular Biology*, 8:1, 2013.

[31] P. E. Compeau, P. A. Pevzner, and G. Tesler. How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology*, 29(11):987–991, 2011.

[32] A. Cooper, C. Lalueza-Fox, S. Anderson, A. Rambaut, J. Austin, and R. Ward. Complete mitochondrial genome sequences of two extinct moas clarify ratite evolution. *Nature*, 409(6821):704–707, 2001.

[33] A. Cooper and H. N. Poinar. Ancient DNA: Do It Right or Not at All. *Science*, 289(5482):1139–1139, 2000.

[34] M. Csűrös. How to infer ancestral genome features by parsimony: Dynamic programming over an evolutionary tree. In *Models and Algorithms for Genome Evolution*, 29–45. Springer, 2013.

[35] A. C. Darling, B. Mau, F. R. Blattner, and N. T. Perna. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Research*, 14(7):1394–1403, 2004.

[36] A. E. Darling, I. Miklós, and M. A. Ragan. Dynamics of genome rearrangement in bacterial populations. *PLoS Genetics*, 4(7):e1000 128, 2008.

[37] R. Darwin Charles. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life. *Murray, London*, 1859.

[38] W. H. Day, D. S. Johnson, and D. Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, 81(1):33–42, 1986.

[39] F. Denoeud, L. Carretero-Paulet, A. Dereeper, G. Droc, R. Guyot, M. Pietrella, C. Zheng, A. Alberti, F. Anthony, *et al.* The coffee genome provides insight into the convergent evolution of caffeine biosynthesis. *Science*, 345(6201):1181–1184, 2014.

[40] V. Deshpande, E. D. Fung, S. Pham, and V. Bafna. Cerulean: A hybrid assembly using high throughput short and long reads. In *International Workshop on Algorithms in Bioinformatics*, 349–363. Springer, 2013.

[41] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[42] T. Dobzhansky and A. H. Sturtevant. Inversions in the Chromosomes of Drosophila Pseudoobscura. *Genetics*, 23(1):28–64, 1938.

[43] M. Drancourt and D. Raoult. Palaeomicrobiology: current issues and perspectives. *Nature Reviews Microbiology*, 3(1):23–35, 2005.

[44] S. R. Eddy. A new generation of homology search tools based on probabilistic inference. In *Genome Informatics*, vol. 23, 205–211. 2009.

[45] J. S. Farris. Phylogenetic analysis under Dollo's Law. *Systematic Biology*, 26(1):77–88, 1977.

[46] P. Feijão and J. Meidanis. SCJ: A Breakpoint-Like Distance that Simplifies Several Rearrangement Problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(5):1318–1329, 2011.

[47] P. Feijão. Reconstruction of ancestral gene orders using intermediate genomes. *BMC Bioinformatics*, 16(Suppl 14):S3, 2015.

[48] P. Feijão and E. Araújo. Fast ancestral gene order reconstruction of genomes with unequal gene content. *BMC Bioinformatics*, 17(14):187, 2016.

[49] M. Feldman, M. Harbeck, M. Keller, M. A. Spyrou, A. Rott, B. Trautmann, H. C. Scholz, B. Päffgen, J. Peters, *et al.* A high-coverage Yersinia pestis Genome from a 6th-century Justinianic Plague Victim. *Molecular Biology and Evolution*, msw170, 2016.

[50] J. Felsenstein. Numerical methods for inferring evolutionary trees. *Quarterly Review of Biology*, 379–404, 1982.

[51] J. Felsenstein. Parsimony in systematics: biological and statistical issues. *Annual Review of Ecology and Systematics*, 14:313–333, 1983.

[52] J. Felsenstein. *Inferring phylogenies*, vol. 2. Sinauer Associates Sunderland, 2004.

[53] G. Fertin. *Combinatorics of Genome Rearrangements*. MIT Press, 2009.

[54] W. Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Biol.*, 20:406–416, 1971.

[55] S. Gnerre, I. MacCallum, D. Przybylski, F. J. Ribeiro, J. N. Burton, B. J. Walker, T. Sharpe, G. Hall, T. P. Shea, *et al.* High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences of the United States of America*, 108(4):1513–1518, 2011.

[56] S. Goodwin, J. D. McPherson, and W. R. McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[57] W. Haak, I. Lazaridis, N. Patterson, N. Rohland, S. Mallick, B. Llamas, G. Brandt, S. Nordenfelt, E. Harney, *et al.* Massive migration from the steppe was a source for Indo-European languages in Europe. *Nature*, 522(7555):207–211, 2015.

[58] E. Hagelberg, M. Hofreiter, and C. Keyser. Ancient DNA: the first three decades. *Philosophical Transactions of the Royal Society B*, 370:20130 371, 2015.

[59] E. Haghshenas, F. Hach, S. C. Sahinalp, and C. Chauve. CoLoRMap: Correcting Long Reads by Mapping short reads. *Bioinformatics*, 32(17):i545–i551, 2016.

[60] J. A. Hartigan. Minimum Mutation Fits to a Given Tree. *Biometrics*, 29(1):53, 1973.

[61] R. Higuchi, B. Bowman, M. Freiberger, O. A. Ryder, and A. C. Wilson. DNA sequences from the quagga, an extinct member of the horse family. *Nature*, 312(5991):282–284, 1984.

[62] M. Hofreiter, J. L. A. Paijmans, H. Goodchild, C. F. Speller, A. Barlow, G. G. Fortes, J. A. Thomas, A. Ludwig, and M. J. Collins. The Future of Ancient DNA: Technical Advances and Conceptual Shifts. *Bioessays*, 37:284–293, 2015.

[63] M. Hofreiter, D. Serre, H. N. Poinar, M. Kuch, and S. Pääbo. Ancient DNA. *Nature Reviews Genetics*, 2(5):353–359, 2001.

[64] G. Holley, R. Wittler, and J. Stoye. Bloom Filter Trie: an alignment-free and reference-free data structure for pan-genome storage. *Algorithms for Molecular Biology*, 11(1):1, 2016.

[65] F. Hu, J. Zhou, L. Zhou, and J. Tang. Probabilistic Reconstruction of Ancestral Gene Orders with Insertions and Deletions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(4):667–672, 2014.

[66] P. Husemann and J. Stoye. Phylogenetic comparative assembly. *Algorithms for Molecular Biology*, 5(3), 2010.

[67] P. Husemann and J. Stoye. r2cat: synteny plots and comparative assembly. *Bioinformatics*, 26(4):570–571, 2010.

[68] B. R. Jones, A. Rajaraman, E. Tannier, and C. Chauve. ANGES: reconstructing ANcestral GEnomeS maps. *Bioinformatics*, 28(18):2388–2390, 2012.

[69] J. Kim, D. M. Larkin, Q. Cai, Asan, Y. Zhang, R.-L. Ge, L. Auvil, B. Capitanu, G. Zhang, *et al.* Reference-assisted chromosome assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 110(5):1785–1790, 2013.

[70] J. Kováč, B. Brejová, and T. Vinař. A practical algorithm for ancestral rearrangement reconstruction. In *International Workshop on Algorithms in Bioinformatics*, 163–174. Springer, 2011.

[71] M. Krings, A. Stone, R. W. Schmitz, H. Krainitzki, M. Stoneking, and S. Pääbo. Neandertal DNA sequences and the origin of modern humans. *Cell*, 90(1):19–30, 1997.

[72] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: an information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009.

[73] S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biology*, 5:R12, 2004.

[74] W. J. Le Quesne. The uniquely evolved character concept and its cladistic application. *Systematic Biology*, 23(4):513–517, 1974.

[75] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25:1754–1760, 2009.

[76] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, *et al.* The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[77] N. Luhmann, C. Chauve, J. Stoye, and R. Wittler. Scaffolding of ancient contigs and ancestral reconstruction in a phylogenetic framework. In *Brazilian Symposium on Bioinformatics*, 135–143. Springer, 2014.

[78] N. Luhmann, D. Doerr, and C. Chauve. Improved assemblies and comparison of two ancient Yersinia pestis genomes. *bioRxiv*, 073445, 2016.

[79] N. Luhmann, M. Lafond, A. Thévenin, A. Ouangraoua, R. Wittler, and C. Chauve. The SCJ Small Parsimony Problem for Weighted Gene Adjacencies. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2017.

[80] N. Luhmann, A. Thévenin, A. Ouangraoua, R. Wittler, and C. Chauve. The SCJ small parsimony problem for weighted gene adjacencies. In *International Symposium on Bioinformatics Research and Applications*, 200–210. Springer, 2016.

[81] J. Ma. A probabilistic framework for inferring ancestral genomic orders. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2010*, 179–184. 2010.

[82] J. Ma, A. Ratan, B. J. Raney, B. B. Suh, W. Miller, and D. Haussler. The infinite sites model of genome evolution. *Proceedings of the National Academy of Sciences of the United States of America*, 105(38):14 254–14 261, 2008.

[83] J. Ma, A. Ratan, B. J. Raney, B. B. Suh, L. Zhang, W. Miller, and D. Haussler. DUPCAR: Reconstructing Contiguous Ancestral Regions with Duplications. *Journal of Computational Biology*, 15(8):1007–1027, 2008.

[84] J. Ma, L. Zhang, B. B. Suh, B. J. Raney, R. C. Burhans, W. J. Kent, M. Blanchette, D. Haussler, and W. Miller. Reconstructing contiguous regions of an ancestral genome. *Genome Research*, 16(12):1557–1565, 2006.

[85] T. Magoč and S. L. Salzberg. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, 27(21):2957–2963, 2011.

[86] J. Mahillon and M. Chandler. Insertion sequences. *Microbiology and Molecular Biology Reviews*, 62(3):725–774, 1998.

[87] I. Mandric and A. Zelikovsky. ScaffMatch: scaffolding algorithm based on maximum weight matching. *Bioinformatics*, 31(16):2632–2638, 2015.

[88] J. Maňuch, M. Patterson, R. Wittler, C. Chauve, and E. Tannier. Linearization of ancestral multichromosomal genomes. *BMC Bioinformatics*, 13(Suppl 19):S11, 2012.

[89] S. Marciniak, J. Klunk, A. Devault, J. Enk, and H. N. Poinar. Ancient human genomics: the methodology behind reconstructing evolutionary pathways. *Journal of Human Evolution*, 79:21–34, 2015.

[90] M. Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):pp–10, 2011.

[91] A. McNally, N. R. Thomson, S. Reuter, and B. W. Wren. 'Add, stir and reduce': Yersinia spp. as model bacteria for pathogen evolution. *Nature Reviews Microbiology*, 14(3):177–190, 2016.

[92] I. Miklós and H. Smith. Sampling and counting genome rearrangement scenarios. *BMC Bioinformatics*, 16(Suppl 14):S6, 2015.

[93] W. Miller, D. I. Drautz, A. Ratan, B. Pusey, J. Qi, A. M. Lesk, L. P. Tomsho, M. D. Packard, F. Zhao, *et al.* Sequencing the nuclear genome of the extinct woolly mammoth. *Nature*, 456(7220):387–390, 2008.

[94] R. Ming, R. VanBuren, C. M. Wai, H. Tang, M. C. Schatz, J. E. Bowers, E. Lyons, M.-L. Wang, J. Chen, *et al.* The pineapple genome and the evolution of CAM photosynthesis. *Nature genetics*, 47(12):1435–1442, 2015.

[95] I. Minkin, A. Patel, M. Kolmogorov, N. Vyahhi, and S. Pham. Sibelia: a scalable and comprehensive synteny block generation tool for closely related microbial genomes. In *International Workshop on Algorithms in Bioinformatics*, 215–229. Springer, 2013.

[96] M. Muffato and H. R. Crollius. Paleogenomics in vertebrates, or the recovery of lost genomes from the mist of time. *BioEssays*, 30(2):122–134, 2008.

[97] F. Murat, A. Louis, F. Maumus, A. Armero, R. Cooke, H. Quesneville, H. R. Crollius, and J. Salse. Understanding Brassicaceae evolution through ancestral genome reconstruction. *Genome biology*, 16(1):1, 2015.

[98] N. Nagarajan, C. Cook, M. Di Bonaventura, H. Ge, A. Richards, K. A. Bishop-Lilly, R. DeSalle, T. D. Read, and M. Pop. Finishing genomes with limited resources: lessons from an ensemble of microbial genomes. *BMC Genomics*, 11(1):242, 2010.

[99] N. Nagarajan and M. Pop. Sequence assembly demystified. *Nature Reviews Genetics*, 14(3):157–167, 2013.

[100] Y. Nakatani, H. Takeda, Y. Kohara, and S. Morishita. Reconstruction of the vertebrate ancestral genome reveals dynamic genome reorganization in early vertebrates. *Genome Research*, 17(9):1254–1265, 2007.

[101] D. E. Neafsey, R. M. Waterhouse, M. R. Abai, S. S. Aganezov, M. A. Alekseyev, J. E. Allen, J. Amon, B. Arca, P. Arensburger, *et al.* Highly evolvable malaria vectors: The genomes of 16 Anopheles mosquitoes. *Science*, 347(6217):1258 522–1258 522, 2015.

[102] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[103] L. Orlando, A. Ginolhac, M. Raghavan, J. Vilstrup, M. Rasmussen, K. Magnussen, K. E. Steinmann, P. Kapranov, J. F. Thompson, *et al.* True single-molecule DNA sequencing of a pleistocene horse bone. *Genome Research*, 21(10):1705–1719, 2011.

[104] L. Orlando, A. Ginolhac, G. Zhang, D. Froese, A. Albrechtsen, M. Stiller, M. Schubert, E. Cappellini, B. Petersen, *et al.* Recalibrating Equus evolution using the genome sequence of an early Middle Pleistocene horse. *Nature*, 499(7456):74–78, 2013.

[105] A. Ouangraoua and M. Raffinot. On the Identification of Conflicting Contiguities in Ancestral Genome Reconstruction. *Journal of Computational Biology*, 21(1):64–79, 2014.

[106] A. Ouangraoua, E. Tannier, and C. Chauve. Reconstructing the architecture of the ancestral amniote genome. *Bioinformatics*, 27(19):2664–2671, 2011.

[107] S. Pääbo. Ancient DNA: extraction, characterization, molecular cloning, and enzymatic amplification. *Proceedings of the National Academy of Sciences of the United States of America*, 86(6):1939–1943, 1989.

[108] S. Pääbo, H. Poinar, D. Serre, V. Jaenicke-Després, J. Hebler, N. Rohland, M. Kuch, J. Krause, L. Vigilant, and M. Hofreiter. Genetic Analyses from Ancient DNA. *Annual Review of Genetics*, 38(1):645–679, 2004.

[109] B. Paten, D. R. Zerbino, G. Hickey, and D. Haussler. A unifying model of genome evolution under parsimony. *BMC Bioinformatics*, 15(1):206, 2014.

[110] M. Patterson, G. Szöll\Hosi, V. Daubin, and E. Tannier. Lateral gene transfer, rearrangement, reconciliation. *BMC Bioinformatics*, 14 Suppl 15:S4, 2013.

[111] L. Pauling and E. Zuckerkandl. Chemical paleogenetics. *Acta Chemica Scandinavica*, 17:S9–S16, 1963.

[112] D. Paulino, R. L. Warren, B. P. Vandervalk, A. Raymond, S. D. Jackman, and I. Birol. Sealer: a scalable gap-closing application for finishing draft genomes. *BMC Bioinformatics*, 16:230, 2015.

[113] A. Peltzer, G. Jäger, A. Herbig, A. Seitz, C. Kniep, J. Krause, and K. Nieselt. EAGER: efficient ancient genome reconstruction. *Genome Biology*, 17(1), 2016.

[114] Penel, S and Arigon, A-M and Dufayard, J-F and Sertier, A-S and Daubin, V and Duret, L and Gouy, M and Perrière, G. Databases of homologous gene families for comparative genomics. *BMC Bioinformatics*, 10(S3), 2009.

[115] P. Pevzner. *Computational molecular biology: an algorithmic approach*. MIT press, 2000.

[116] H. N. Poinar, C. Schwarz, J. Qi, B. Shapiro, R. D. E. Macphee, B. Buigues, A. Tikhonov, D. H. Huson, L. P. Tomsho, *et al.* Metagenomics to paleogenomics: large-scale sequencing of mammoth DNA. *Science*, 311(5759):392–394, 2006.

[117] A. Rajaraman, E. Tannier, and C. Chauve. FPSAC: fast phylogenetic scaffolding of ancient contigs. *Bioinformatics*, 29(23):2987–2994, 2013.

[118] A. Rajaraman, J. Zanetti, J. Maňuch, and C. Chauve. Algorithms and complexity results for genome mapping problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[119] S. Rasmussen, M. E. Allentoft, K. Nielsen, L. Orlando, M. Sikora, K.-G. Sjögren, A. G. Pedersen, M. Schubert, A. Van Dam, *et al.* Early Divergent Strains of Yersinia pestis in Eurasia 5,000 Years Ago. *Cell*, 163(3):571–582, 2015.

[120] K. Sahlin, F. Vezzi, and others. BESST - Efficient scaffolding of large fragmented assemblies. *BMC Bioinformatics*, 15:281, 2014.

[121] L. Salmela, K. Sahlin, V. Mäkinen, and A. I. Tomescu. Gap filling as exact path length problem. In *Research in Computational Molecular Biology*, 281–292. Springer, 2015.

[122] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):909–917, 1999.

[123] D. Sankoff and M. Blanchette. The median problem for breakpoints in comparative genomics. In *International Computing and Combinatorics Conference*, 251–263. Springer, 1997.

[124] D. Sankoff and J. H. Nadeau. Chromosome rearrangements in evolution: From gene order to genome sequence and back. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11 188–11 189, 2003.

[125] D. Sankoff and P. Rousseau. Locating the vertices of a Steiner tree in an arbitrary metric space. *Mathematical Programming*, 9(1):240–246, 1975.

[126] B. V. Schmid, U. Büntgen, W. R. Easterday, C. Ginzler, L. Walløe, B. Bramanti, and N. C. Stenseth. Climate-driven introduction of the Black Death and successive plague reintroductions into Europe. *Proceedings of the National Academy of Sciences of the United States of America*, 112(10):3020–3025, 2015.

[127] M. Schubert, L. Ermini, C. Sarkissian, H. Jónson, A. Ginolhac, R. Schaefer, *et al.* Characterization of ancient and modern genomes by SNP detection and phylogenomic and metagenomic analysis using PALEOMIX. *Nature Protocols*, 9:1056–1082, 2014.

[128] M. Schubert, A. Ginolhac, S. Lindgreen, J. F. Thompson, K. A. AL-Rasheid, E. Willerslev, A. Krogh, and L. Orlando. Improving ancient DNA read mapping against modern reference genomes. *BMC Genomics*, 13(1):178, 2012.

[129] M. Semeria, E. Tannier, and L. Guéguen. Probabilistic modeling of the evolution of gene synteny within reconciled phylogenies. *BMC Bioinformatics*, 16 Suppl 14:S5, 2015.

[130] P. Siguier, J. Filée, and M. Chandler. Insertion sequences in prokaryotic genomes. *Current Opinion in Microbiology*, 9(5):526–531, 2006.

[131] D. Simon and B. Larget. Bayesian Analysis to Describe Genomic Evolution by Rearrangement (BADGER), version 1.02 beta. *Department of Mathematics and Computer Science, Duquesne University*, 2004.

[132] J. T. Simpson and M. Pop. The theory and practice of genome sequence assembly. *Annual review of genomics and human genetics*, 16:153–172, 2015.

[133] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. Jones, and I. Birol. ABySS: a parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.

[134] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[135] M. A. Spyrou, R. I. Tukhbatova, M. Feldman, J. Drath, S. Kacki, J. Beltrán de Heredia, S. Arnold, A. G. Sitdikov, D. Castex, *et al.* Historical Y. pestis Genomes Reveal the European Black Death as the Source of Ancient and Modern Plague Pandemics. *Cell Host & Microbe*, 19(6):874–881, 2016.

[136] M. Steel and D. Penny. Parsimony, likelihood, and the role of models in molecular phylogenetics. *Molecular Biology and Evolution*, 17(6):839–850, 2000.

[137] M. Stiller, G. Baryshnikov, H. Bocherens, A. G. d'Anglade, B. Hilpert, S. C. Münzel, R. Pinhasi, G. Rabeder, W. Rosendahl, *et al.* Withering away – 25,000 years of genetic decline preceded cave bear extinction. *Molecular biology and evolution*, 27(5):975–978, 2010.

[138] J. Stoye and R. Wittler. A unified approach for reconstructing ancient gene clusters. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(3):387–400, 2009.

[139] A. H. Sturtevant. A Case of Rearrangement of Genes in Drosophila. *Proceedings of the National Academy of Sciences of the United States of America*, 7(8):235–237, 1921.

[140] E. Tannier, C. Zheng, and D. Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10(1):120, 2009.

[141] W. K. Thomas, S. Pääbo, F. X. Villablanca, and A. C. Wilson. Spatial and temporal continuity of kangaroo rat populations shown by sequencing mitochondrial DNA from museum specimens. *Journal of Molecular Evolution*, 31(2):101–112, 1990.

[142] T. J. Treangen and S. L. Salzberg. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nature Reviews Genetics*, 13(1):36–46, 2012.

[143] T. M. Tumpey, A. García-Sastre, J. K. Taubenberger, P. Palese, D. E. Swayne, and C. F. Basler. Pathogenicity and immunogenicity of influenza viruses with genes from the 1918 pandemic virus. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):3166–3171, 2004.

[144] G. H. Van Domselaar, P. Stothard, S. Shrivastava, J. A. Cruz, A. Guo, X. Dong, P. Lu, D. Szafron, R. Greiner, and D. S. Wishart. BASys: a web server for automated bacterial genome annotation. *Nucleic Acids Research*, 33(suppl 2):W455–W459, 2005.

[145] M. Višnovská, T. Vinar, and B. Brejová. DNA Sequence Segmentation Based on Local Similarity. In *ITAT 2013 Proceedings*,, 36–43. 2013.

[146] D. M. Wagner, J. Klunk, M. Harbeck, A. Devault, N. Waglechner, J. W. Sahl, J. Enk, D. N. Birdsell, M. Kuch, *et al.* Yersinia pestis and the Plague of Justinian 541 − 543 AD: a genomic analysis. *The Lancet Infectious Diseases*, 14(4):319–326, 2014.

[147] B. J. Walker, T. Abeel, T. Shea, M. Priest, A. Abouelliel, S. Sakthikumar, C. A. Cuomo, Q. Zeng, J. Wortman, *et al.* Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS One*, 9(11):e112 963, 2014.

[148] E. Willerslev, A. J. Hansen, R. Rønn, T. B. Brand, I. Barnes, C. Wiuf, D. Gilichinsky, D. Mitchell, and A. Cooper. Long-term persistence of bacterial DNA. *Current Biology*, 14(1):R9–R10, 2004.

[149] R. Wittler. *Phylogeny-based Analysis of Gene Clusters*. Ph.D. Thesis, Faculty of Technology, Bielefeld University, 2010.

[150] R. Wittler, J. Maňuch, M. Patterson, and J. Stoye. Consistency of sequence-based gene clusters. *Journal of Computational Biology*, 18(9):1023–1039, 2011.

[151] A. W. Xu and B. M. E. Moret. GASTS: Parsimony Scoring under Rearrangements. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, *et al.*, eds., *Algorithms in Bioinformatics*, vol. 6833, 351–363. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[152] S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.

[153] D. R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829, 2008.

[154] H. Zhao and G. Bourque. Recovering genome rearrangements in the mammalian phylogeny. *Genome Research*, 19(5):934–942, 2009.

[155] C. Zheng. Pathgroups, a dynamic data structure for genome reconstruction problems. *Bioinformatics*, 26(13):1587–1594, 2010.

[156] C. Zheng and D. Sankoff. On the Pathgroups approach to rapid small phylogeny. *BMC Bioinformatics*, 12(Suppl 1):S4, 2011.

[157] D. L. Zimbler, J. A. Schroeder, J. L. Eddy, and W. W. Lathem. Early emergence of Yersinia pestis as a severe respiratory pathogen. *Nature Communications*, 6:7487, 2015.