A Novel *Pseudo*-Alignment Approach to Fast Genomic Sequence Comparison

Chris-André Leimeister

Institut für Mikrobiologie und Genetik
Abteilung für Bioinformatik

Master’s Thesis
of the Center for Computational Sciences
at the Georg-August-Universität Göttingen

01. Oktober 2015
Georg-August-Universität Göttingen
Institute of Computer Science

Goldschmidtstraße 7
37077 Göttingen
Germany

☎ +49 (551) 39-172000
✉ +49 (551) 39-14403
✉ office@informatik.uni-goettingen.de
☎ www.informatik.uni-goettingen.de

First Supervisor: Prof. Dr. Burkhard Morgenstern
Second Supervisor: Dr. Peter Meinicke
I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, 01. Oktober 2015
Abstract

Standard methods for sequence analysis and phylogeny reconstruction are based on (multiple) sequence alignments. These methods are known to be accurate but if larger genomic sequences are to be analysed they reach their limits. Consequently, faster but less precise alignment-free methods are increasingly used for genomic sequence analysis. In this work, a novel approach to fast genomic sequence comparison was developed. This method combines the advantages of alignment-free and alignment-based methods. A key component of this approach are so called spaced-words which contain don’t care or wildcard symbols at certain pre-defined positions. Don’t care positions of spaced-word matches between two genomic sequences are used to determine the so called pseudo-alignment of a pair of sequences. To elude background matches a simple yet powerful filter technique was proposed. An efficient and highly parallelizable implementation of the pseudo-alignment approach, including the filter technique, was developed and implemented in C++. To evaluate the performance of the pseudo-alignment approach artificially generated sequences as well as real word genomes were used. Tests showed that the pseudo-alignment is competitive with other state of the art methods and even outperform them on some data sets.
Contents

1 Introduction 1
   1.1 Motivation, Problem Statement and Goal ........................................ 2
   1.2 Structure ..................................................................................... 3

2 Fundamentals 5
   2.1 Notation ..................................................................................... 5
   2.2 Alignment-Free Methods .............................................................. 6
      2.2.1 Taxonomy ............................................................................. 6
      2.2.2 Feature Frequency Profiles (FFP) ........................................... 8
      2.2.3 Composition Vector Tree (CVTree) ....................................... 9
      2.2.4 Return Time Distribution (RTD) ........................................... 10
      2.2.5 Frequency Chaos Game Representation (FCGR) ....................... 11
      2.2.6 CO-Phylog .......................................................................... 13
      2.2.7 Spaced-Words and Evolutionary Distances ......................... 15
      2.2.8 The Average Common Substring Approach (ACS) .................. 19
      2.2.9 Estimation of Evolutionary Distances (Kr) ......................... 21
      2.2.10 Andi .................................................................................. 23
   2.3 Genome Alignment Methods ....................................................... 26

3 Method 29
   3.1 The Pseudo-Alignment Approach ............................................... 30
   3.2 A Filter Technique to Discard Background Matches ..................... 32
   3.3 Implementation ........................................................................... 36

4 Benchmark 39
   4.1 Data Sets and Evaluation Procedure ............................................ 39
   4.2 Results .......................................................................................... 40

5 Conclusion and Outlook 49

References 50
Chapter 1

Introduction

While 'biology has traditionally been an observational rather than a deductive science' [30], the emergence of modern genetics has changed the face of biology. The exponentially accumulating amount of molecular data available today require computational methods and probabilistic models to understand the data. This has led to the rapidly expanding field of modern bioinformatics or computational biology. Bioinformatic tools have become essential for modern biological data analyses and the demand for faster and/or more accurate methods is greater than ever.

Although computers have been used for biological analyses since the early 1960s, computational biology or bioinformatics was still in its infancy. While some authors pointed out the importance of computational biology at that time [6], the gold rush started during the 1990s and accelerated as modern DNA sequencing emerged. These modern or next-generation sequencing technologies produce sequence data with an unprecedented throughput at very low costs. A good example to emphasize the dimension of the data volume generated is the 1000 Genomes Project [2] which was an international research effort with the goal to sequence the genomes of a large number of people to determine genetic variations among humans. This project started in 2008 and was declared completed in 2011 as more than 1000 complete human genomes were sequenced and published. In January 2014 the next milestone was reached: The long-awaited $1,000 genome has arrived which means that it is now possible to sequence an entire human genome for less than 1,000 USD. Having such fast and cheap sequencing technologies available, scientists recently concluded that 'genomics is Big Data science and is going to get much bigger, very soon.' [50]. By doubling of the data volume every 7 months, it is expected that genomic sequence data might exceed other Big Data domains as YouTube, Twitter and even astronomy in the future [50].

With the rapid accumulation of sequenced genomes of different species, one of the most fundamental goal in biology is closer than ever before: The reconstruction of the history of life on earth. This history shows evolutionary relationships among various species depicted as a tree, called evolutionary or phylogenetic tree. Since all living organisms on earth descended from one last
universal ancestor the history of life can be shown as a phylogenetic tree, called the tree of life. To reconstruct the phylogeny of a set of species it is necessary to determine when the organisms diverged because organisms which are more closely related share a more recent common ancestor than organisms that are more distantly related. Traditionally, this was done by comparing differences or similarities in their physical characteristics. Nowadays, however, species are usually compared on a molecular level, i.e. genetic variations among organisms are investigated. Genetic variations, also called mutations or substitutions, are changes in the DNA, the hereditary material of life. The more mutations happened between two species the more distantly they are related, the fewer mutations happened, the closer they are related. The most common method to identify such mutations is to calculate a sequence alignment of homologous genes or proteins which can also contain insertions and deletions of single or multiple nucleotides/amino acids. Standard methods as maximum likelihood can be applied to the sequence alignment to infer phylogenies. Such methods are called character based methods because a sequence alignment reveals which nucleotides changed into another. This allows to distinguish between mutations so that different substitution models can be used as JC [24], F81 [14], GTR [51] and others. Another class of phylogenetic methods are distance based. These methods define distances between all pairs of sequences, forming a \( (m^2) \) distance matrix for a set of \( m \) sequences. While character based methods are known to be very accurate their run time is quite high. Distance based methods, however, are much faster but have a considerably lower accuracy. Additionally, cluster algorithms as neighbor-joining [43] or UPGMA [49] must be applied to the distance matrix to obtain a tree.

In this work, a novel approach was developed for faster and more accurate sequence comparison and phylogeny reconstruction.

1.1 Motivation, Problem Statement and Goal

Traditionally, comparative sequence analysis for phylogenetic inference are based on pairwise or multiple sequence alignments. However, with thousands of complete sequenced genomes available in databases, these methods reach their limits. While alignment-based methods can be applied to a set of homologous genes or proteins, to calculate an optimal global alignment of a set whole genomes is far from reality because prokaryotic and, especially, eukaryotic genomes are far too large. Even an optimal pairwise alignment takes time proportional to the product of the sequence lengths which already exceeds the computational power of modern computers. In addition to being slow, genomic events as rearrangements, repeats, horizontal gene transfer and partial relatedness among a group of organisms hamper a meaningful sequence alignment. Consequently, alignment-free methods are increasingly used for comparative genomic analyses since they circumvent the problematic alignment step.

Most alignment-free methods define a distances either by comparing the word composition of
sequences or by comparing substring matches among sequences. While most distances defined in this way are heuristics which are not based on an explicit model of molecular evolution some authors developed methods that are able to estimate substitution rates between unaligned sequences. Although these methods work very well on simulated data, their performance varies considerably when applied to different real-world data sets. For some data sets the results are remarkably good but applied to other data sets they perform poorly. A comprehensive summary and a critical assessment on modern alignment-free methods is given in Chapter 2.

While alignment-based methods face notorious difficulties when applied to real world genomes, alignment-free methods can alleviate these problems. However, as pointed out, alignment-free methods are substantially less accurate. Therefore, the goal of this thesis is to develop an alignment-like approach which utilizes properties of alignment-free methods but circumvent the drawbacks of alignment-based methods. The input of the method is a pair of unaligned sequences and the output is a set of columns which represent the homologous nucleotides among the sequences. The homologous nucleotides are written above each other like in an alignment but the positions of the columns are ignored, thus the set of all columns is called the pairwise pseudocalignment of the sequences.

1.2 Structure

- Chapter 1 - General introduction to the topic, including motivation, problem statement and the goal of the thesis. The general idea of the developed approach is outlined.

- Chapter 2 - A taxonomy of modern alignment-free methods is proposed and selected approaches are described and analysed. This chapter is closed by a brief introduction to genome alignment programs.

- Chapter 3 - Describes the method formally and introduces a filter technique which allows to discard background matches. Additionally an efficient implementation is presented.

- Chapter 4 - Evaluation of the pseudocalignment approach and comparison to other alignment-free methods. Real-word data sets as well as artificial data sets are used as benchmark.

- Chapter 5 - This chapter concludes the thesis and gives a short outlook how this project can be continued.
Chapter 2

Fundamentals

In this chapter, an overview of modern approaches to large scale genomic sequence analysis is given. After the basic notation is introduced, a taxonomy of modern alignment-free methods is proposed. Selected methods of this taxonomy will be summarized and critically analysed by pointing out their strengths and weaknesses. Finally, this chapter is closed by a brief introduction to genome alignment approaches.

2.1 Notation

This section follows the terminology introduced by \[28\] \[29\]. For an alphabet \(\Sigma\) and \(\ell \in \mathbb{N}\), \(\Sigma^\ell\) denotes the set of all sequences or contiguous words over \(\Sigma\) of length \(\ell\). In a biological context, the alphabet \(\Sigma\) represents a set of nucleotides or amino acids, respectively. In this work, the focus is on genomic sequence comparison, therefore the alphabet is restricted to \(\Sigma = \{A, C, G, T, N\}\), whereas A, C, G, T are the nucleotides and N denotes characters that are either not yet sequenced or the correct nucleotide could not be determined. In genomic sequences, also other nucleotides occur which are not in \(\Sigma\), but in this work it will be assumed that such nucleotides are converted to Ns. For a sequence \(S\) over \(\Sigma\), \(S[i]\) is the \(i\)-th element of \(S\) and \(S[i..j]\) with \(i \leq j\) is denoted as the contiguous substring of \(S\), starting at \(i\) and ending at \(j\). The length of a substring is denoted as \(|S|\) and \(S[i..|S|]\) is denoted as the \(i\)-th suffix of \(S\). Some alignment-free methods are based on so called spaced words instead of contiguous words. A spaced-word \(w\) with integers \(\ell\) and \(k\) with \(k \leq \ell\) is defined as a pair \((w', P)\) where \(w' \in \Sigma^k\) is a contiguous word of length \(k\) and \(P \in \{0, 1\}^\ell\) is a sequence of 0 and 1 characters of length \(\ell\), such that there are exactly \(k\) positions \(i\) in \(P\) with \(P[i] = 1\). \(P\) is called the underlying pattern of \(w\) and \(k\) is called the weight of \(P\). A spaced word \((w', P)\) with matching positions \(p_1 < p_2 < \cdots < p_k\) in pattern \(P\) occurs at position \(i\) in \(S\), if \(S[i + p_j - 1] = w'[j]\) for all \(1 \leq j \leq k\). This notation of spaced seeds (words) was firstly introduced by \[33\].
2.2 Alignment-Free Methods

In this section, modern alignment-free methods are outlined to introduce common techniques for genomic comparisons. By pointing out their limitations, the need for a novel sequence comparison method will be highlighted.

2.2.1 Taxonomy

During the last two decades various methods and tools have been proposed to calculate distances between sequences. While the most exciting approaches were published in the last 5-10 years, scientists already developed methods and ideas to compare sequences before the emergence of modern DNA sequencing. A highly cited paper which review alignment-free methods proposed up to the year 2002 can be found here [55]. While several authors reviewed modern alignment-free methods [18, 54], a taxonomy of modern approaches is to my knowledge not existent. Therefore, a suggestion for a taxonomy is given in figure 2.1 However, a summary of all methods proposed in the last decades is out of the scope of this work, thus the taxonomy is restricted to the most exciting methods which introduce new concepts and ideas.

Alignment-free methods can be broadly divided into three classes: Methods which are based on k-mers, i.e. of words of a fixed length $k$, methods which defined distances based on common substring matches and methods which are based on information theory. However, the boundaries between the classes are blurry and hybrid methods often exist.

The majority of alignment-free methods are based on word counts, as FFP [46, 47, 48] and CVTree [39]. That is, overlapping word frequencies of length $k$ are determined for each sequence, leading to $|\Sigma|^k$ dimensional vectors. Distances between pairs of sequences are then defined as the distance between the corresponding frequency vectors. A method which compares spaced k-mer (word) compositions instead of contiguous k-mers was proposed recently [5]. This approach was later extended to a multiple spaced word approach [22, 28] and a novel evolutionary distance was proposed to estimate substitutions between pairs of unaligned sequences [35, 36]. This was, however, not the first idea to use spaced word matches in an alignment-free context. The CO-Phylog approach [58] estimates distances based on so called micro-alignments which compare the don’t care positions of spaced-word matches between sequences. Some alignment-free methods are based on the chaos game representation (CGR) [23] technique which is a graphical representation of a genome. For sequence comparison this approach was extended to a frequency chaos game representation (FCGR) and used in alignment-free study of the brassicales clade [17]. The last k-mers based like approach in this section is named return time distribution (RTD) [25]. It defines a distance between sequences based on lengths between equal k-mers within sequences.

The second class of methods are based on substring matches, also called matching statistics,
which can also be regarded as word matches between sequences of variable lengths. One of the first approaches based on this idea was the \textit{average common substring approach (ACS)} \cite{52}. It defines the average length of longest common substrings as measure of similarity and turns it into a symmetric distance measure. This approach was later generalized to the \textit{k-mismatch average common substring approach} \cite{29}. These authors proposed a greedy heuristic to estimate the length longest common substring matches with up to \(k\)-mismatches. Since both methods lack of a model of molecular evolutionary the authors \cite{13,20} proposed an estimator for the number of substitutions between two unaligned sequences based on the length of longest common substring matches. To incorporate mismatches, they later proposed an improved estimator (\textit{andi}) which is based on gap-free local alignments bounded by maximal unique matches \cite{19}. The section of methods based substring matches is closed by mentioning the \textit{underlying subwords approach} \cite{7}. The basic idea of this approach is to eliminate overlapping substring matches. The intention behind this idea is not to distort the distance by repetitive but not evolutionary relevant repeats. However, since the implementation provided by the authors is quite slow and therefore the benefit of this method is
uncertain because it can not be applied to larger genomes, a detailed analysis of this approach will be omitted in the next chapter.

Information theory [44] has a wide range of applications in biology, see [53] for an overview, and alignment-free sequence comparison is one field of it. The distances defined by k-mer based and substring based methods are often related or directly based on information theory but, although the boundaries are blurry, information theoretic approaches will be listed in an own class. The motivation for this is, that the goal of modern alignment-free methods tends to estimate mutation rates so that models of molecular evaluation can be used while information theoretic approaches, however, are rather rough heuristic measures. Nevertheless, they have proven to produce meaningful results in an alignment-free context.

There is a close relation between information theory and data compression: The entropy $H$ is the ultimate data compression. In a biological context data compression can be used to estimate distances between sequences: Similar sequences are more compressible than divergent sequences [18]. One popular compression scheme for alignment-free distance estimation is known as Lempel–Ziv factorization [60] which is a dictionary based approach. Sequences are usually decomposed into substrings leading to various methods to decomposition- or grammar-based distance measures [38, 52], whereas a grammar defines a set of rules to decompose a string. Two other information theoretic approaches to alignment-free sequence analysis are the Base base correlation (BBC) [32] and the information correlation and partial information correlation (IC-PIC) [15]. Information theoretic methods will not be analysed because this work is mainly concerned about defining distances based on models of molecular evolution.

### 2.2.2 Feature Frequency Profiles (FFP)

The Feature Frequency Profiles (FFP) approach [46, 47, 48] is a refined version of k-mer counting. A feature (or k-mer) is a word of a certain length and the authors defined the feature profile of a sequence as the collection of all features occurring in the sequence. Differences in these frequency profiles are then used to calculate distance scores between sequences. The underlying hypothesis is that the frequency profiles between closely related organisms are similar. To define a distance between between the frequency profiles they implemented the euclidean distance

$$d_{euclidean}(Q, S) = \sqrt{\sum_{i=0}^{n} (q_i - s_i)^2}$$  \hspace{1cm} (2.1)$$

whereas $Q$ and $S$ are frequency vectors, $q_i$ is the frequency of the $i$-th feature of length $k$ in $Q$ and $s_i$ is defined equally for $S$. They also implemented the Jensen-Shannon divergence [31], whereby $Q$
and $S$ are the relative frequency vectors

$$d_{jensen\text{-}shannon}(Q, S) = \frac{1}{2} KL(Q, M) + \frac{1}{2} KL(S, M)$$  \hspace{1cm} (2.2)$$

where

$$M = \frac{1}{2} (Q + S)$$  \hspace{1cm} (2.3)$$

is the mean of $Q$ and $S$, and $KL$ is is the Kullback–Leibler divergence [26], defined as

$$KL(Q, M) = \sum_{i=0}^{\infty} \log_2 \left( \frac{q_i}{m_i} \right) q_i$$  \hspace{1cm} (2.4)$$

However, both distances are clearly not derived from an explicit model of molecular evolution but nevertheless generate reasonable phylogenetic trees. Why these rough distance measures work at all will be discussed in section 2.2.7.

Two modifications were suggested to improve the performance of FFP: **RY-coding** and **block-FFP**. **Block-FFP** is used if two genomes differ significantly in their length. Then the larger genome is divided into blocks of the length of the shorter genome and the frequency profiles of these blocks are compared to the shorter genome. **RY-coding** is used to reduce the DNA alphabet from four nucleotides to two nucleotides by combining the two purines (A and G) to $R$ and the two pyrimidines (C and T) to $Y$. Their main argument for this reduction was the lower memory requirement, because there are only $2^k$ possible words if the alphabet is reduced compared to $4^k$ possible words if the standard alphabet is used.

### 2.2.3 Composition Vector Tree (CVTree)

The basic idea of the **Composition Vector Tree (CVTree)** approach [39] is to define a distances based on (relative) $k$-mer frequencies, similar to FFP, but additionally, these authors proposed to subtract a random background from the frequency vectors. To do so, they denoted the frequency of a $k$-mer as $f(\alpha_1\alpha_2\ldots\alpha_k)$, whereas $\alpha_1\alpha_2\ldots\alpha_k$ is defined as a $k$-mer. Next, they denoted the probability of the appearance of a $k$-mer $\alpha_1\alpha_2\ldots\alpha_k$ in a sequence as $p(\alpha_1\alpha_2\ldots\alpha_k)$, which is the relative frequency of the $k$-mer. They predicted the appearance of a $k$-mer based on a markovian assumption, as follows

$$p^0(\alpha_1\alpha_2\ldots\alpha_k) = \frac{p(\alpha_1\alpha_2\ldots\alpha_{k-1}) - p(\alpha_2\alpha_3\ldots\alpha_k)}{p(\alpha_2\alpha_3\ldots\alpha_{k-1})}$$  \hspace{1cm} (2.5)$$
To form a composition vector for a sequence $Q$, the components are computed as

$$q(\alpha_1 \alpha_2 \ldots \alpha_k) = \begin{cases} \frac{p(\alpha_1 \alpha_2 \ldots \alpha_k) - p^0(\alpha_1 \alpha_2 \ldots \alpha_k)}{p(\alpha_1 \alpha_2 \ldots \alpha_k)}, & p \neq 0 \\ 0, & p = 0 \end{cases}$$

(2.6)

and the composition vector of a sequence is denoted as $Q = (q_1, q_2, \ldots, q_n)$. To calculate distances, they defined a correlation $C(Q, S)$ between two composition vectors as the cosine angle in the $N$-dimensional space of composition vectors

$$C(Q, S) = \frac{\sum_{i=0}^{n} q_i \cdot s_i}{\sqrt{\sum_{i=0}^{n} q_i^2 \cdot \sum_{i=0}^{n} s_i^2}}$$

(2.7)

This correlation is turned into a distance by

$$d_{CVTree}(Q, S) = \frac{1 - C(Q, S)}{2}$$

(2.8)

The underlying idea to subtract frequencies expected by chance alone was based on the idea to compare sequence compositions that are evolutionary relevant. However, the main drawback is that it is still a rough measure of dissimilarity, similar to $FFP$. Consequently, these $k$-mer based distances are only designed to recover the topology of a tree rather than the branch lengths. Branch lengths of phylogenetic trees should reflect the time since two organisms diverged.

### 2.2.4 Return Time Distribution (RTD)

The Return Time Distribution (RTD) [25] measures the number of nucleotides between equal $k$-mers within a sequence. In contrast to methods which defines distances based on frequency vectors, RTD calculates distances based on statistical parameters. To define the RTD approach formally, two words in sequence $S$, starting at $i$ and $j$ are considered: Word $w_i = S[i..i+k]$ and word $w_j = S[j..j+k]$ with $w_i = w_j$ and $i < j$. If for all words $w_r = S[r..r+k]$ in $S$ with $i < r < j$ holds that $w_i \neq w_r \neq w_j$, then the return time between $w_i$ and $w_j$ is defined as $j - i - 1$. For each $k$-mer the return time as well as the frequencies of the return times are determined. Then, the mean ($\mu$) and the standard deviation ($\sigma$) of these frequencies are calculated, leading to pairs of $(\mu, \sigma)$, the return time distribution. An example is shown in figure 2.2 which illustrates the procedure for one $k$-mer of length one, i.e. a single nucleotide. Finally, a distance between two sequences is defined as

$$d_{RTD}(Q, S) = \sqrt{\sum_{i=0}^{n} (q_{i\mu} - s_{i\mu})^2 + (q_{i\sigma} - s_{i\sigma})^2}$$

(2.9)

whereas $q_{i\mu}$ refers to the $i$-th return time distribution for sequence $Q$, and $q_{i\sigma}$ is defined accordingly.
2.2. ALIGNMENT-FREE METHODS

This new distance was applied to various virus genomes and the results indicate that this approach lead to reasonable phylogenetic trees. However, the RTD approach does not calculate real evolutionary distance but one can say that if only few mutations happened between two sequences, \( \mu \) and \( \sigma \) should not differ much. For two diverged organisms however, the distributions should differ considerably. Yet, meaningful branch length of the recovered tree cannot be expected using this method.

2.2.5 Frequency Chaos Game Representation (FCGR)

Frequency Chaos Game Representation (FCGR) \cite{17,23,56} is a graphical representation of k-mer frequencies. Distances based on FCGR do not differ from other word count based methods, but yet, it is an interesting approach to visualize word compositions of whole genomes. To generate a chaos game representation for a DNA sequence, at first an empty \( 1 \times 1 \) square is drawn and each corner is labelled with a nucleotide. In most studies C is placed in the upper left, G upper right, T lower right and A in the lower left. The start point is the center of the square, i.e. (0.5,0.5). Then, starting with the first nucleotide in sequence S, the first dot is plotted in the middle between the starting point (0.5,0.5) and the corner which represents the first nucleotide. This procedure is subsequently repeated by plotting new dots in between the previous dot and the corresponding corner until the end of the sequence is reached. This algorithm can be described by following equations.
\[ GCR_0 = (0.5, 0.5) \]
\[ GCR_i = \begin{cases} 
   CGR_{i-1} + 0.5 \times (CGR_{i-1} + (0.0, 0.0)), & seq_i = C \\
   CGR_{i-1} + 0.5 \times (CGR_{i-1} + (1.0, 0.0)), & seq_i = G \\
   CGR_{i-1} + 0.5 \times (CGR_{i-1} + (0.0, 1.0)), & seq_i = A \\
   CGR_{i-1} + 0.5 \times (CGR_{i-1} + (1.0, 1.0)), & seq_i = T 
\end{cases} \] (2.10)

An example of this procedure is shown in figure 2.3a. This results in a chaos game representation of a sequence, whereas similar genomes result in similar plots. To make a connection between chaos game representation and \(k\)-mer frequencies, the square is divided into subsquares which then represent the \(k\)-mer frequencies. For example: If dinucleotid (2-mer) frequencies are to be determined, the grid is divided into a \(4 \times 4\) grid which leads to 16 subsquares. Each subsquare represent one of the 16 dinucleotids (AA, AC, AG,..., TG, TT). The frequencies of the dinucleotids are shown in different grey scales. The more frequent one dinucleotid is, the darker the corresponding subsquare becomes. In this way, the frequencies of all \(k\)-mers can be determined by dividing the scare into a \(2^k \times 2^k\) grid. The division into subsquares for trinucleotids is shown in figure 2.3b. Therefore, the division of a chaos game representation of a sequence into a \(2^k \times 2^k\) grid is called the frequency chaos game representation of a genome. Various distances can be used to define a distance between two FCGRs as the Euclidean distance or the Pearson distance. Both distances were also used in the phylogenetic analysis of the brassicales clade [17]. These authors also plotted the FCGRs of various genomes and three of these images are shown in figure 2.4. The images show similar structures between the different genomes. However, the only new contribution of FCGRs is the visual representation of the \(k\)-mer frequencies and therefore the distances calculated based on FCGRs remain the same as for the standard \(k\)-mer distances.

![Figure 2.3](image-url)

Figure 2.3: a) Procedure to generate a chaos game representation for the sequence GCAC T. b) Procedure to determine trinucleotide frequencies by dividing the CGR map into a \(2^3 \times 2^3\) grid. [17]
2.2. ALIGNMENT-FREE METHODS

Figure 2.4: FCGRs \((k = 8)\) of three different whole genomes: A) *Brassica rapa*, B) *Arabidopsis thaliana* and C) *Citrus clementina* [17].

2.2.6 CO-Phylog

CO-Phylog [58] was the first method which used spaced words instead of contiguous words. To introduce the CO-Phylog approach the idea of a spaced word is shortly recap: For two integer \(\ell\) and \(k\) with \(\ell \geq k\), a spaced word \(w = (w', P)\) of length \(\ell\) with \(P \in \{0, 1\}^{\ell}\) and match positions at \(p_1 < p_2 < \cdots < p_k\) in \(P\), occurs in a sequence \(S\) at position \(i\), if \(S[i + p_j - 1] = w'[j]\) for all \(1 \leq j \leq k\).

For the CO-Phylog approach the nucleotides at the don’t care are compared, therefore for a spaced word \(w = (w', P)\) in \(S\), starting at position \(i\) of length \(\ell \geq k\) and \(P \in \{0, 1\}^{\ell}\) let \(r_1 < r_2 < \cdots < r_{\ell-k}\) be the don’t care positions (the 0s) in pattern \(P\) and let \(o'\) be the word occurring at \(S[i + r_j - 1]\) for all \(1 \leq j \leq \ell - k\). The authors called a spaced word \(w\) in \(S\) a context if and only if for all \(w_i = w'_i\) holds that \(o'_i = o'\). The nucleotides at the don’t care positions, \(o'\), are called the object. As example consider the sequence \(S = AATATTATA\) and the pattern 101: The spaced word at position \(i = 1\) and at position \(i = 6\) with \(w' = AA\) has the object \(o' = T\) for both positions and is therefore a context. However, the spaced word occurring at position \(i = 0\) and \(i = 3\) with \(w' = AT\) is not a context, because for the word at position \(i = 0\) the object is \(o' = A\) and for the word at position \(i = 3\) the object is \(o' = T\).

The set of all contexts for a sequence \(S\) is denoted as \(C(S)\). To define a distance between two sequences they calculated the sets of contexts for each sequence and then compared the object of each context to each other. Formally, for two sequences \(S\) and \(Q\) they defined

\[
I_i = \begin{cases} 
0, & \text{if } o' \text{ of } c_i(S) = o' \text{ of } c_i(Q) \\
1, & \text{if } o' \text{ of } c_i(S) \neq o' \text{ of } c_i(Q) 
\end{cases} 
\]  

and

\[ R = |C(S) \cap C(Q)| \]
Finally, the CO-Phylog distance is defined as

\[ d_{CO}(Q, S) = \frac{\sum_{i=0}^{R} I_i}{R} \]  

(2.13)

Tests on real word genomes have shown that the CO-Phylog approach produces remarkable good results for very closely related genomes. To find an explanation for the good performance, a closer look will be taken at the distances. To do so, a spaced word match will be defined first. For two sequences \( S \) and \( Q \) and a pattern \( P \) a spaced word match occurs between \( S \) and \( Q \) at position \( i \) in \( S \) and \( j \) in \( Q \) if \( S[i + p_r - 1] = Q[j + p_r - 1] \) holds for all \( 1 \leq r \leq k \), whereas \( p_r \) are the positions of the 1s in the underlying pattern. The first observation is that if the weight of a pattern is sufficiently large word matches which occur just by change are very rare. Therefore, the majority of word matches between sequences can be considered as homologous matches. Most patterns used in the study of the authors contained one single don’t care position, so that only one nucleotide (object) was compared to another. If one assumes that mostly objects of homologous matches are compared, then the distance calculated by CO-Phylog can be regarded as an estimator for the \( p \)-distance. The \( p \)-distance is the observed or uncorrected number of mismatches between two sequences. The number of mismatches observed (or estimated) between two sequences do not necessary reflect the real number of mutations that happened because some nucleotides might have changed back. For example, a nucleotide A could have changed to C and then back to A and no substitution would

![Figure 2.5](image)

Figure 2.5: Distances are shown as a function of the number of substitutions per site. The green squares are distances calculated by CO-Phylog, the red triangles show the correct distances and the blue crosses refers to the \( p \)-distance
be visible in an alignment even though two substitutions happened.

To turn the observed mismatch rate into an evolutionary distance which also accounts for back substitution a simple correction, called Jukes Cantor (JC) \[24\] was proposed already in 1969. The JC model assumes that each nucleotide changes into another with equal rate. To analyse the distances calculated by CO-Phylog a pair of sequences with a known number of substitutions was simulated, using the JC model. The distances calculated by CO-Phylog should underestimate the real number of substitutions and the \( p \)-distance should be calculated. However, as can be seen in figure 2.5 the distances calculated are not the \( p \)-distances. It rather looks like the number of substitutions are estimated even though no model of molecular evolution is used to correct the distances. My view is, that the CO-Phylog distance is rather a poor estimator of the \( p \)-distance but coincidentally estimates the number of substitutions up to a substitution rate of 0.7 quite well. However, I also want to point out that this statement is only true for larger substitution rates. As can be seen in figure 2.5 for substitution rates up to 0.2 the \( p \)-distance is approximated quite well. The good performance on closely related organisms can be explained by the marginal differences between the \( p \)-distance and the corrected distances if only few mismatches occurred.

The CO-Phylog approach introduced a novel idea to compare don’t care positions of spaced word matches between a pair of sequences. While the authors turned this into a distance it can be also seen as a character based method, since nucleotides can compared directly. The pseudo-alignment approach developed in this work does not turn the object comparison into a distance but into an alignment-like structure, so that character-based methods can be applied to it.

### 2.2.7 Spaced-Words and Evolutionary Distances

In contrast to CO-Phylog which defines a distance based on the don’t care positions of spaced word matches between sequences, the spaced words approach \[5\] defines a distance based on spaced word frequencies. That is, for a sequence \( S \) the spaced word frequencies are counted, which means words with contain wild card positions at the don’t care positions in the underlying pattern. A distance between two spaced word frequency vectors are then defined as the distance between the sequences, similar to FFP and other generic \( k \)-mer/\( k \)-word methods. The motivation for using spaced words instead of contiguous words is that spaced words are statistically less dependent on each other.

The idea of spaced seeds (words) was originally proposed by \[33\] who applied it for database searching. They showed that spaced seeds (words) in this context are far superior compared to contiguous seeds (words) because they are more likely to find a match in homologous regions between sequences. Therefore, spaced seeds (words) are also used in the well know local alignment program BLAST \[3\]. While the general idea of spaced words is not new, it was just recently proposed to use it in an alignment-free context to define distances between sequences \[5\]. The results of this study showed that spaced words can improve the performance but it depends on the underlying pattern. This approach was later extended to a multiple spaced words approach \[28\], where instead of one pattern
A set of patterns $P = \{P_1, P_2, \ldots, P_m\}$ was used. The distance between a pair of sequences was defined as the average of the distances defined by the patterns $P \in P$, i.e.

$$d_{\text{spaced}}(Q, S) = \frac{1}{m} \sum_{P \in P} d_P(Q, S)$$

(2.14)

whereas for $d_P(Q, S)$ the euclidean distance $d_1$ and the Jensen-Shannon divergence $d_2$ were used.

The evaluation of the multiple pattern version showed that this extension improved the results significantly. But why do distances calculated based on the euclidean distance or based on the Jensen-Shannon divergence lead to biologically meaningful results? To answer this question a new distance metric introduced by [35, 36] will be summarized first.

Generally, spaced word matches can either occur due to homology or just by chance. To define an estimator for the number of substitutions per site, two sequences of length $L$ are considered. Additionally, for simplicity it will be assumed that the sequences have a match probability $p$, no insertions or deletions took place and the sequences are aligned, i.e. homologous regions stand above each other. Therefore, the match probabilities for single nucleotides can be expressed as

$$P(S[i] = Q[i]) = \begin{cases} p & \text{for } i = j \\ q & \text{for } i \neq j \end{cases}$$

(2.15)

whereas $p$ is the probability of a homologous match and $q$ is the probability of a background match. Consequently, the probability of a spaced word match of length $k$ is $p^k$ for a homologous match and $q^k$ for a background match. An example is shown in figure 2.6.

Based on this model and for a set of $m$ pattern, each of length $\ell$ and weight $k$, the expected value can

\[ E(N) = m \left( L - \ell + 1 \right) \left( 1 - p \right) k + \left( L - \ell \right) \left( L - \ell + 1 \right) \left( \frac{1}{4} \right) k \]


Figure 2.6: Illustration of word matches between two sequences $Q$ and $S$. The green block shows a homologous word match while the red block refers to a random match.
be expressed as

\[ E(N) = m \cdot \left[ (L - \ell + 1) \cdot p^k + (L - \ell + 1) \cdot (L - \ell) \cdot q^k \right] \]

(2.16)

For each position in Q there is exactly one homologous position in the other sequence and for each position in Q there is a chance of a random match at any position in the other sequence with the background probability. Therefore, the \( p \)-distance can be estimated based on the number of word matches between two sequences

\[ \hat{p} = \frac{L}{m \cdot (L - \ell + 1) - (\ell - L) \cdot q^k} \]

(2.17)

Together with Jukes Cantor this estimator can be turned into an evolutionary distance \( d_{EV} \). To assess the accuracy of the estimator, a pair of sequences with a known number of substitutions was generated and the results are shown in figure 2.7. On these simulated data, the estimator works very well and is able to estimate distances very accurately up to a substitution rate of 0.8, then the results start to vary which can be seen by the error bars.

Although this distance measure works well on artificially generated sequences, real world genomes are far more complex. One major difficulty is that real genomes consist of highly repetitive regions. The number of word matches is calculated as the dot product of the frequency vectors and if highly repetitive words occur the distance is distorted. To address this problem a binary count

![Figure 2.7: The estimated distance based on spaced word matches is shown as a function of the number of substitutions per site.](image-url)
was implemented so that word matches are only counted once. Another problem for real world data sets is that homologies between two genomic sequences can be located on different strands, which means a sequence $Q$ can have a homologous match in the reverse strand of $S$. Therefore, the distance $d_{REC}^{REV}$ for homologous matches in the reverse strand and the distance $d_{bin}^{REV}$ for the binary count version were implemented.

Now the question why the *euclidean* distance and the *Jensen-Shannon* divergence lead to reasonable phylogenies can be answered: If the weight of the pattern is *sufficiently* large then most matches can be regarded as homologous matches between the sequences. Therefore, the frequency vectors mainly consists of 1s and 0s, i.e. either there is a homologous match or the is none. Thus the *Jensen-Shannon* distance can be approximated by calculating $N - L$ and the *euclidean* as the square root of it $\sqrt{N - L}$.

While this new distance measure looks very promising the performance on real world data sets was not as good as one would assume based on the simulated data. One drawback is that insertions and deletions happen during evolution and therefore the sequences differ in length. It is therefore unknown which parts of the sequences are homologous and which are not. The authors proposed to take the shorter sequence length as $L$ in the equation but this is rather a rough measure of the homologous length between sequences. However, the authors showed that if applied to sequences with insertions and deletions the distances are just slightly overestimated. To explain this, the number of word matches are plotted as a function of the mismatch rate $(1 - p)$ in figure 2.8. As can be seen, there is no linear relationship between the number of word matches and the mismatch rate. This has an influence of the estimated distance if the mismatch rate between two genomes varies. For simplicity, let’s assume that for two sequences of length $L$ one half of the sequences are highly conserved, e.g. the mismatch rate is 0.01 and the other half of the sequences mutated much faster, e.g. with a mismatch rate of 0.1 then the average mismatch rate should be 0.055. However, because of the non linear relationship between between the number of word matches and the mismatch rate the number of mismatches are underestimated. The estimated mismatch rate based on the number of word matches would be about 0.042. This scenario is shown in figure 2.8 whereas the black lines show the two different mismatch rates for the sequences, the red cross shows the estimated mismatch rate and the green cross shows the correct mismatch rate. Therefore, the estimator is distorted if two sequences share homologies over the whole genome but with different mutation rates. However, the positive aspect of this observation is that insertions and deletion do not have a large impact if the sequences are relatively closely related because there are overwhelmingly many matches within the homologous part such that the distances are only slightly overestimated, as shown in the study [35].
Figure 2.8: The graph shows the expected number of spaced word matches as a function of the mismatch rate \((1 - p)\). The two black lines indicate different mismatch rates within two sequences. The green cross shows the correct average mismatch rate but the red cross shows the estimated mismatch rate based on the number of spaced word matches.

### 2.2.8 The Average Common Substring Approach (ACS)

The Average Common Substring Approach (ACS)\(^{52}\) was one of the first approaches which defines a distance based on longest common substring matches between two sequences. Such longest common substring matches can be interpreted as word matches of variable length. While computing word matches among sequences is quite easy if the length of the words are fixed, it becomes much more complex to determine word matches of variable length. Therefore, more complex data structures as suffix trees\(^ {57}\) or suffix arrays\(^ {34}\) must be used to achieve a linear run time which is necessary for large scale genomic comparison. To define a distance between two sequences \(Q\) and \(S\), the ACS approach determines for each position \(i\) in \(Q\) the length \(s_Q(i)\) of the longest substring starting at position \(i\) in \(Q\) that exactly matches some substring in the other sequence \(S\). The average of these
lengths are then defined as a similarity measure

\[ L(Q, S) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} s_Q(i) \]  

This measure of similarity is turned into a distance by

\[ d(Q, S) = \frac{\log(S)}{L(Q, S)} - \frac{\log(Q)}{L(Q, Q)} \]  

However, this distance is not symmetric. Sources of asymmetry can be for instance unequal sequence lengths or varying number of copies of a certain genetic element. Both examples are illustrated in figure 2.9. To obtain a symmetric distance the authors defined their distance as the average of both asymmetric values

\[ d(Q, S)_{\text{ACS}} = \frac{L(Q, S) + L(S, Q)}{2} \]  

Simultaneously, a similar approach for genome comparison was proposed by [21], called shortest unique substrings (shustrings). A shustring is defined as a substring of length \( x \) such that \( S[i..i+x-1] \) is unique while \( S[i..i+x-2] \) is not. In this approach they determined all shustrings for each position \( i \) within one sequence which can be regarded as a measure of repetitiveness. They also derived an estimator for the expected shustring length for random sequences. Later, they extended this approach where they determined for each position \( i \) in \( Q \) the shustring that is absent from another sequence \( S \) [20]. Shustrings and longest common substrings are almost identical because if a longest common substring is extended by one position to the right then it coincide with the shustring. Therefor, the lengths of shustrings is equivalent to the length of longest common substrings plus one.

A generalization of the ACS approach, the k-mismatch average common substring approach (kmacs), was proposed recently [29]. Instead of calculating lengths of exact substring matches they used substring matches with up to \( k \) mismatches. To do so, the authors defined \( s^k_Q(i) \) as the length of

![Figures (see separate files)]

**Fig. 1:** Sources of asymmetry in the average shustring length, \( \ell(Q, S) \). A: Sequences \( S_1 \) and \( S_2 \) differ in length and as a result share only local homology (—), in which case \( \ell(S_1, S_2) < \ell(S_2, S_1) \); B: \( S_1 \) contains a lower copy number of a particular genetic element (box) than \( S_2 \), in which case again \( \ell(S_1, S_2) < \ell(S_2, S_1) \).

**Fig. 2.9:** A: Sequence \( Q \) is shorter than sequence \( S \) and therefore, they share only local homologies. B: The genetic element in \( Q \), shown as a box, occurs multiple times in sequence \( S \) which can happen due to gene duplication. Both cases lead to asymmetric values of average common substring lengths, i.e. \( L(Q, S) \neq L(S, Q) \) [20].
the longest substring of $Q$ starting at position $i$ that matches some substring of $S$ with up to $k$ mismatches, minus $k$. Then they defined a distance based on the average length of the $k$-mismatch substrings similar to the ACS approach. However, calculating substrings with $k$-mismatches takes even more time than an optimal pairwise sequence alignment, therefore the $k$-mismatch substrings were approximated by using a greedy heuristic. The first step of this heuristic is to determine for position $i$ in $Q$ the longest common substring as in the ACS approach. To incorporate $k$-mismatches this exact match is extended to the right. In detail, let $j$ be the start position in $S$ that matches a longest common substring starting at position $i$ in $Q$. Then, this initial substring match is extended from position $i + s_Q(i)$ in $Q$ and position $j + s_Q(j)$ in $S$ until all $k$-mismatches are used up. However, the corresponding position $j$ in $S$ that is the longest exact match to a substring starting at $i$ in $Q$ is maybe not unique. Therefore, all positions $j$ in $S$ that match a substring starting at $i$ are extended to determine the approximated length of the longest common substring with up to $k$-mismatches (minus $k$).

Tests showed that this greedy heuristic reduced the run time significantly, so that this methods can be used to analyse large genomic sequences. It was also shown that the extension of the longest common substrings can improve the results significantly if the sequences are relatively divergent. Compared to the original ACS approach the user needs to define how many mismatches should be used. The requirement of an user defined parameter is always a draback but results showed that increasing the value of $k$ generally improves the performance and for larger values the results tends to converge.

The distance defined by ACS and also used for kmacs depends on the substitution rate between sequences but this measure is far away from an evolutionary distance. To address this problem an estimator for the number of substitutions based on the length of longest common substrings/shustrings was proposed by [20] and will be described in the next section in detail.

### 2.2.9 Estimation of Evolutionary Distances (Kr)

Longest common substring or shustrings among two sequences can be interrupted by a point mutation or by insertions/deletions. Therefore, the more mutations or insertions/deletions happened between two sequences the shorter are the average shustring lengths. If one ignores insertions/deletions then the number of substitutions can be estimated by the average length of the shustrings. Such an estimator was proposed by [20] and implemented in the program Kr. In detail, if one assumes that $m$ mutations happened between two sequences $Q$ and $S$ then the probability to have a shustring match of length $x$ is given as $(1 - \frac{m}{|Q|})^x$. Therefore, the probability that a shustring of length $X$ is longer than some threshold $t$ can be expressed as

$$P(X > t) = (1 - \frac{m}{|Q|})^x \approx e^{-tm/|Q|}$$

(2.21)
However, this is only correct if one assumes that all shustring matches are due to common homology. But this is far from truth because the chance of background matches grow with increasing divergence. Therefore, the authors corrected it for background matches as follows

\[
P(X \leq t) = (1 - e^{-tm/|Q|})(1 - 4^{-t}|Q|)
\]  \hspace{1cm} (2.22)

whereas is was assumed that all nucleotides occur with an equal rate. Since this is an estimator for the number of mismatches between two sequences it must be corrected to obtain an evolutionary distance, similar to the estimator described in section 2.2.7, by using the Jukes Cantor model \[24\]. The authors showed that this estimator approximates the rate of substitution per site very well up to 0.5 substitutions per site, as can be seen in figure 2.10.

Figure 2.10: Estimated distances based on average shustring lengths are shown as a function of the number of substitutions per site (K) \[20\].

The criticism of this estimator is essentially the same as for the estimator based on spaced word matches. This estimator does not account for insertions and deletion and the average shustring length as a function of the substitution rate is exponentially distributed as can be seen in figure 2.11 and also directly from the equation 2.21. Therefore, this distance estimator should also be influenced if mutations among sequences are not equally distributed.
2.2. ALIGNMENT-FREE METHODS

Figure 2.11: The average shustring length is shown as a function of the number of substitutions per site ($K$) [20].

### 2.2.10 Andi

The most recently proposed method is called andi [19]. It can be regarded as a hybrid version of the estimator $K_r$ and $kmacs$. The motivation for this approach was to improve the estimator $K_r$ by taking mismatches into account but to circumvent the a user specific parameter as for $kmacs$. To do so, they looked for mismatches between two exact matches which they called anchor points. Anchor points are selected based on three criteria: Uniqueness, equidistance and a minimum length of the anchor points. The first criteria, uniqueness, refers to maximal unique matches between two sequences $Q$ and $S$. The basic idea of maximal unique matches was originally proposed for the genome alignment approach MUMmer [27]. A maximal unique matches between $S$ and $Q$ is only used as an anchor points if it has a minimum length. A minimum length is necessary to avoid anchor points that only occurred by chance instead of common ancestry. The minimum anchor length is defined as the 97.5% quantile of the distribution $X_i^*$, whereas

$$P(X_i^* \leq x) = \sum_{k=0}^{x} 2^k \left( \begin{array}{c} x \\ k \end{array} \right) p^k (1/2 - p)^{x-k} (1 - p)^k (1 - p)^{x-k} |S|$$

(2.23)

and $X_i^*$ is the length of a match starting at position $i$ in $Q$ and matching a substring at position in $S$. The equidistance criteria is based on equal segment lengths. That is, the number of nucleotides between two anchor points in $Q$ must be equal to number of nucleotides between two anchor points.
in \( S \). An example is shown in figure 2.12. The idea behind that is if the length between two anchor points differ, insertions or deletions must have happened and therefore, these segments are not homologous. For all pairs of anchor points which fulfil all three criteria, the mismatches between them are counted, leading to an estimated mismatch rate. However, this mismatch count is asymmetrical and therefore the average of both mismatch rates is taken. Finally, this mismatch rate is turned into an evolutionary distance by using the Jukes and Cantor correction [24]. Tests on simulated data as well as on real word genomes showed that this new distance measure improved the results significantly. This estimator approximates the number of substitutions very well up to a substitution rate of about 0.5 as can be seen in figure 2.13. For larger substitution rates the new distance fails.

Figure 2.12: Anchor points are shown as blue and green blocks. A: Equally spaced anchor points; mismatches between these blocks are counted. B: Unequally spaced anchor points; they will be discarded [19].

Figure 2.13: Estimated distances are shown as a function of the number of substitutions per site (\( K \)) [19].
Counting the mismatches between anchor points circumvents the problem of unevenly distributed mismatches compared to their previous approach Kr or compared to the distance estimator based on the number of spaced word matches. However, one drawback of this approach is that only unique matches are taken but there might be evolutionary relevant gene duplications which would be lost. Therefore, this distance can be interpreted as an alignment of a selection of homologous regions within two sequences. This is already very close to a genome alignment and the benefit of andi is the far superior run time compared to genome alignment programs.
2.3 Genome Alignment Methods

Alignments of whole genomes are in general very difficult tasks because of genomic events as rearrangements and duplications of genetic elements. The greatest challenge, however, is the enormous size of an entire genome. Therefore, an optimal alignment is far from reality and heuristic methods are required to identify homologies among sequences. The first approach that tackled this problem was based on maximal unique matches (MUMs), known as MUMmer \[11\]. To align homologous regions of two genomes the MUMmer algorithm works in three steps. In the first step all MUMs between both sequences are identified. These MUMs are then sorted to extract the longest possible set of matches which occur in the same order of both genomes. The last step is to close the gaps between the ordered matches by local identification of indels or small mutated regions etc. Multiple improvements (versions) were published since then, e.g. version 2.0 \[12\] in 2002 and version 3.0 \[27\] in 2004. In version 3.0 the authors extended their approach such that non-unique repetitive matches alongside unique matches can be identified as well. A visualization of MUMs is shown in figure 2.14.

Figure 2.14: Dotplot of the set of all MUMmers between \textit{helicobacter pylori} J99 (y-axis) and \textit{helicobacter pylori} 26695 (x-axis). MUMs found between forward strands are plotted as red dots while green dots show reverse MUMs \[1\].

A generalization of maximal unique matches was developed by \[10\] and implemented in the program \textit{mauve}. Instead of finding MUMs in a pair of sequences the authors proposed to look for multi-
MUMs, i.e MUMs that occur among a set of genomes. These Multi-MUMs are then used to calculate a guide tree. In the next step, a subsets of multi-MUMs are used as anchors and are partitioned into collinear groups. Then, a recursive anchoring technique is used to identify additional alignment anchors within and outside the collinear groups. Finally, a progressive alignment is performed for each collinear group according to the guide tree. A visualization of a genome alignment calculated with mauve is shown in figure 2.15.

One of the more recent proposed genome alignment program is called mugsy. Mugsy is built on the foundation of the MUMmer approach which is used to calculate all pairwise alignments for a set of genomic sequences. Then a graph based algorithm is used to identify local collinear blocks which are then aligned progressively using SeqAn::TCoffee.

Although many different algorithms have been developed over the two decades a (multiple) sequence alignment of whole genomes remains difficult. While modern genome aligners are able to align prokaryotic sequences in acceptable time, they reach their limits when applied to eukaryotic genomes which are several magnitudes larger in size. A genome alignment is, however, not only an algorithmic problem but the challenge also underlie in the nature of the evolution. Horizontal gene transfer and evolutionary relevant repeats interfere often meaningful results.
Chapter 3

Method

Methods that rely on spaced word matches between two genomic sequences have proven to produce good results. However, as pointed out in the introduction, current approaches which define distances based on these matches have multiple drawbacks. The spaced words substitution rate estimator \[35,36\] faces problems if mutations are not equally distributed among the sequences or if the sequences share local homologies only. Since the homologous regions between two genomes are not known, this estimator cannot be used for sequences where huge insertions or deletions happened.

CO-Phylog \[58\] circumvent this problem but performs inferior if larger distances are to be estimated. This approach usually incorporates one single don’t care position in the pattern so that the \(p\)-distance can be estimated. However, using patterns with only one don’t care position is statistically more dependent on the previous words as if more don’t care were used. Therefore, one can argue that more conserved regions are compared to each other, which could distort the distances. Moreover, a spaced word match is only considered if it is a context and if not it is discarded. With increasing number of background matches the number of non contexts will increase which distorts the distance. Based on this observation, it becomes evident that patterns with higher weight are required to reduce the number of background matches. Larger weights, however, lead to much fewer word matches so that only a small selection of homologous regions are compared. Additionally, CO-Phylog is unable to take repetitive regions among two genomes into account, because only the object of contexts is compared regardless of how many word matches occurred among the sequences.

In this work it is proposed to calculate a so called pseudo-alignment of two genomic sequences based on spaced word matches. In the section 3.1 the basic approach is described while in the following section 3.2 a filter technique is proposed to reduce background matches.
3.1 The Pseudo-Alignment Approach

To calculate a pairwise pseudo-alignment, all spaced word matches among two sequences are determined and then the don’t care positions are further investigated. The don’t care positions can either be matches or mismatches. A spaced word match with matching and mismatching don’t care positions is shown in figure 4.1. In contrast to CO-Phylog which compares single don’t care positions of unique spaced word matches, the pseudo-alignment approach is designed to work with multiple spaced word matches as well as with patterns containing multiple don’t care positions. To do so, a counting technique is proposed to handle multiple or overlapping spaced word matches. Similar to an alignment, the desired outcome of the pseudo-alignment are columns which represent homologous nucleotides. Yet, in contrast to an alignment, the positions are completely ignored to reduces the space and time complexity. Therefore, this approach is denoted as pseudo-alignment.

The first step of the pseudo-alignment algorithm is to take one sequence as the reference sequence. Let sequence $S$ be the reference sequence and let the spaced word match $w_i$ starting at position $i$ in $S$, with don’t care positions $o_1 < o_2 < o_{\ell-k}$ in pattern $P$, match a word $w_j$ starting at position $j$ in sequence $Q$. Then, the nucleotides occurring at the don’t care positions in $Q$ of the word match are aligned according to the positions $i + o_h - 1$ for all $1 < h < \ell - k$, i.e. aligned to the nucleotides occurring at the don’t care positions in $S$. This procedure to align spaced word match between $S$ and $Q$ according to the reference sequence $S$ is shown in figure 3.2. Since there might be multiple and/or overlapping spaced word matches, multiple (different) nucleotides might be aligned to a position in $S$. For example, a position $i$ in $S$ could have been twice a don’t care position of a spaced word match between $S$ and $Q$. Therefore, two nucleotides would be assigned to this position which could either be the same or different. To determine which nucleotide is correct, i.e. the homologous nucleotide, the frequencies of the nucleotides occurring at the don’t care position of a spaced word match in $Q$ are counted for each position $i$ in $S$ and the most frequent character is considered to be correct. Let $\overline{Q}$ be a sequence of length $|S|$ and the most frequent characters are assigned to $\overline{Q}[i]$ for all positions $i$ in $S$. The basic idea behind this approach is that one can assume to have generally more homologous spaced word matches between two sequences than background matches. However, is is possible that no spaced word match was found for one position or that the frequency count is ambiguous, which means there are two (or more) nucleotide counts that are equal and also the most frequent. In both cases, the character ‘-’ is assigned to $\overline{Q}[i]$. After all assignments are
3.1. THE PSEUDO-ALIGNMENT APPROACH

Figure 3.2: Procedure to align spaced word matches among two sequences $S$ and $Q$ to the reference sequence $S$. The green bar (■) in $S$ and the orange bar (■) in $Q$, as well as the violet bar (■) in $S$ and the blue and red bars (■, ■) in $Q$ symbolize spaced word matches which are aligned according to the reference sequence $S$.

done, $\overline{Q}$ represents the aligned nucleotides (or gap characters ‘-’) to all positions in $S$.

One question remains: Which reference sequence should be chosen. One solution could be to take the shorter sequence as the reference sequence as an estimate for the homologous regions because, clearly, there can’t be more homologous parts as the length of the shorter sequence. However, this is a very rough estimate for the length of shared homologies between two sequences. To circumvent this problem both sequences are considered as reference. As result one obtains two pairwise pseudo-alignments. To merge these two pseudo-alignments into one, for each pseudo-alignment all columns are extracted, whereby columns which consists of ‘-’, called singletons, are discarded. For example, one columns pair is $(A, T)$ where the nucleotide ‘A’ occurring in the reference sequence and the nucleotide ‘T’ occurring in $Q$ are aligned. There are in total 16 different pairs of nucleotide combinations and these frequencies are counted for each pseudo-alignment. The final pseudo-alignment is defined as the minimum frequency count for each of the 16 pair combinations between the sequences. The idea behind it is that only pairs which were found in both pseudo-alignments are merged into the final pseudo-alignment. To illustrate the intention of this idea consider the following example: If a sequence $S$ is taken as reference sequence and an insertion happened in this sequence, then more pairs might be found for this inserted region due to random hits or repeats. However, if $Q$ is taken as reference sequence these pairs would not be found at all.
3.2 A Filter Technique to Discard Background Matches

If two genomic sequences are large and share only small homologous regions then background matches can have a negative impact on the pseudo-alignment. The larger a non homologous region is the higher the chance is that a background match will be found in this region. Since there would not be any homologous matches the idea to count the most frequent nucleotide would fail because there might be only one match which is the background match. In such cases non homologous nucleotides would be aligned which would suggest that the two sequences are more distantly related than they are. Therefore, a simple yet powerful filter technique will be proposed in the following.

In genome alignment programs background matches can be identified by calculating an optimal chain of colinear non-overlapping fragments. This is, however, very time and memory consuming.

![Figure 3.3: The distribution of the number of mismatches of spaced word matches occurring at the don’t care positions between two genomic sequences is shown. Both sequences of length $10^7$ were generated randomly with equal nucleotide frequencies, leading to a match rate of $p = 0.25$, or mismatch rate of $(1-p) = 0.75$. The bars show the empirical frequencies of the number of mismatches occurring at the don’t care positions of spaced word matches. The length of the pattern was $\ell = 62$ and the weight was $k = 12$ and therefore 50 don’t care positions occurred the pattern. The red line follows a binomial distribution with success probability 1-$p$=0.75. Success is defined as a mismatch.](image-url)
3.2. A FILTER TECHNIQUE TO DISCARD BACKGROUND MATCHES

and therefore not applicable for the pseudo-alignment approach. Instead of a chaining algorithm the pseudo-alignment approach filters out spaced word matches based on the number of mismatches that occur at the don’t care positions. The ratio of the number of matches and mismatches of the don’t care positions of spaced word matches is binomial distributed and dependent on the mismatch rate among genomes.

To illustrate this, two random sequences were generated with length $10^7$ and equal nucleotide frequencies. Therefore, the match rate is $p = 0.25$ and the mismatch probability of the sequences is $(1 - p) = 0.75$. The distribution of the number of mismatches of spaced word matches of weight $k = 12$ and length $\ell = 62$ is shown in figure 3.3. As can be seen, it follows a binomial distribution where a mismatch is defined as success. This is because it can be regarded as a probabilistic experiment of 50 independent yes/no (mismatch/match) experiments of which number of successes (mismatches) yields probability $(1-p)$.

To show the distribution for related genomic sequences a pair of sequences with match rate $p$

![Figure 3.4: The distribution of the number of mismatches of spaced word matches occurring at the don’t care positions between two genomic sequences is shown. The nucleotide frequencies are equal but in contrast to the experiment shown in 3.3 the sequences are related. The left plot was generated based on sequences with a match rate of $p = 0.95$ and the plot on the right is based on sequences with a match rate of $p = 0.85$. The bars show the empirical frequencies of the number of mismatches occurring at the don’t care positions of homologous spaced word matches. The same parameters were used as in 3.3. The green line follows a binomial distribution with success probability $1 - p = 0.05$ (left) and $1 - p = 0.15$ (right). Success is defined as a mismatch.](image-url)
CHAPTER 3. METHOD

Figure 3.5: Similar to 3.4 but with different match rates between two homologous sequences. The left plot was generated based on sequences with a match rate of \( p = 0.65 \) and the plot on the right is based on sequences with a match rate of \( p = 0.55 \).

were generated. To identify the homologous matches only, the artificially generated sequences are aligned and contain no insertions or deletions. Therefore, background matches could be identified and discarded. In figures 3.4 and 3.5 it can be seen that the number of mismatches of homologous matches follows a binomial distribution with success probability \((1 - p)\), whereby success is defined as a mismatch. Clearly, the more distantly related the sequences are the more mismatches occur at the *don’t care* positions of *spaced word matches*.

In figure 3.6 the homologous as well as the background matches are considered at the same time. Two peaks can be seen clearly and thus it is possible to distinguish homologous matches from background matches. However, if two sequences are too distantly related, the distribution based on homologous matches and the distribution based on background matches overlap and therefore it is not possible anymore to decide if a *spaced word matches* is a homologous match or a background match. Moreover, the ratio between homologous matches and background matches can be seen in figure 3.6. The number of homologous matches falls sharply with increasing divergence.

Since background matches would distort the *pseudo*-alignment significantly, only *spaced word matches* with up to 25 mismatches are considered as homologous matches, whereby patterns with 50 *don’t care* positions are used. This cut off is illustrated in figure 3.7.
3.2. A FILTER TECHNIQUE TO DISCARD BACKGROUND MATCHES

Figure 3.6: The distribution of the number of mismatches of spaced word matches occurring at the don’t care positions between two related genomic sequences is shown. The left plot was generated based on sequences with a match rate of $p = 0.85$ and the plot on the right is based on sequences with a match rate of $p = 0.7$.

Figure 3.7: Binomial distributions for various success probabilities. The vertical line shows the cut off which defines up to how many mismatches a spaced word matches is considered as a homologous match.
3.3 Implementation

The pseudo-alignment approach is intended to work on genomic sequences, including large eukaryotic genomes. Therefore, a fast run time as well as low memory consumption are crucial factors. To achieve both goals, a sorting based approach is proposed since other data structures and methods are not suitable. The reason is that, in contrast to approaches that rely on word frequencies, the pseudo-alignment approach also requires the positions of spaced word matches to align them according to the reference sequence. To do so, the first step of the algorithm is to lexicographically sort all spaced words occurring in both sequences. That is, for a sequence $S$ and pattern $P$ with matching positions $p_1 < p_2 < \cdots < p_k$, the positions of all spaced words are sorted according to the characters $S[i + p_j - 1]$ for any $i \leq j \leq k$ and vice versa for $Q$. As aforementioned, the pseudo-alignment approach focuses on DNA sequences and thus the alphabet $\Sigma$ is restricted to the 4 nucleotides $\Sigma = \{A, C, G, T\}$. However, especially for larger eukaryotic sequences, unsequenced nucleotides often occur, denoted as $N$s or by other characters that are not in the alphabet $\Sigma$. To address this issue, a spaced word $w_i$, starting at position $i$ in $S$ with pattern $P$ and matching positions $p_1 < p_2 < \cdots < p_k$ is ignored if $S[i + p_j - 1] \notin \Sigma$ for any $i \leq j \leq k$. It is sensible to ignore such word matches because if the real nucleotides are unknown it cannot contribute to a meaningful distance estimation between two sequences. After sorting, one obtains a permutation of the indices according to the lexicographical ordering of the corresponding spaced words. Then, the positions of spaced words are classified into buckets whereas equal spaced words end up in the same bucket. Spaced word matches between two sequences $S$ and $Q$ occur if in the same bucket at least one spaced word exist for each sequences. Once the spaced word matches are identified the nucleotides occurring at the don’t care positions can be easily identified and counted according the position of the reference sequence.

To sort the positions of spaced words efficiently, a simple hash function $h$ maps each spaced word of weight $k$ to $2^k$ bits, i.e. $h: \Sigma^k \rightarrow \{0, 1\}^{2^k}$. In detail, to represent all four characters of the DNA alphabet, the hash function maps each nucleotide into two bits ($A = (00)_2$, $C = (01)_2$, $G = (10)_2$, $T = (11)_2$). To map a spaced word into a binary representation, the bits of single nucleotides are concatenated, using the bitwise left-shift operator ($\ll$) and the bitwise OR operator ($\lor$). For example, the word ATTCG will be mapped to $0011110110$. By doing so, up to 32 character can be sorted at once if a 64 bit computer word is used which is much faster then sorting single nucleotides at a time.

To avoid that all positions of both sequences are kept in the memory at the same time, only subsets of spaced words which share a common prefix smaller than the weight $k$ are considered simultaneously. In particular, only spaced words that share a common prefix of length 4 are considered at the same time because it fits perfectly into a 8-bit machine word. This leads to a total of 256 distinct buckets which are then refined into smaller buckets that represent common spaced words. For each of the 256 buckets the algorithm iterates over the sequences to identify
words with a common prefix of length 4 and then maps the complete spaced words to their binary representation. Consequently, only a fraction of all spaced words are considered at once, reducing the memory consumption significantly. Another advantage of this technique is that it can be easily parallelized which is crucial for large scale genome analysis. To parallelize the described algorithm, each of the 256 buckets are sorted independently, so that each bucket can be processed by one thread. While this is a quite efficient parallelization, for each additionally thread more memory is required. By using 256 threads the memory consumption would be the same as if all positions are considered at the same time. Therefore, the user can scale this algorithm based on the hardware available. For parallelization the OpenMP framework was used.

The theoretical time complexity of this algorithm is $O(n^2)$ because within each bucket all positions of one sequence must be streamed against all positions of the other sequence. In the case that only a few or even only one bucket exist, the theoretical upper bound is clearly quadratic. However, for normal sequences, the buckets are roughly evenly large and if the weight of the pattern is sufficiently large then the average run time is tends to $O(n)$. 
Chapter 4

Benchmark

In this chapter the performance of the pseudo-alignment approach will be evaluated. For a competitive analysis it will be compared to other state of the art alignment-free methods. The first section of this chapter presents the data sets that were used. Additionally, the evaluation procedure will be introduced. In the second section, the results for the pseudo-alignment approach as well as for selected alignment-free methods are shown.

4.1 Data Sets and Evaluation Procedure

To evaluate the performance of the pseudo-alignment, two classes of data sets were used: Real-word data sets and artificially created sequences. For the latter one, sequences with a length of 250kb were generated with an average of $d$ substitutions per sequence position. These artificial sequences are homologous over the complete length, i.e. no insertions or deletions happened. As usual, the distances estimated by the pseudo-alignment are shown as a function of the number of substitutions per site. However, sequences without insertions or deletions are far from reality. Whole genomes can differ substantially in length and therefore, they are often only partially related. Even if homologous genes are to be analysed, insertions and deletions occur frequently. Therefore, pairs of sequence that contains insertions and deletions were generated as well and the estimated distances were compared to the real ones. For each position there was a 1% chance for either an insertion or a deletion. If an insertion or deletion happened the length of the gap was randomly chosen between 1 – 50 nucleotides.

In addition to this direct distance comparison, the main part of the evaluation was carried out by phylogeny construction. That is, for a set of evolutionary related sequences, all pairwise distances were calculated and then, the neighbor joining (NJ) [43] was applied to these distances to infer the corresponding phylogeny. To measure the accuracy of the pseudo-alignment approach, the recovered phylogeny was compared to a reference tree that is considered to be reliable. To compare
two phylogenies, the *Robinson-Folds metric* (RF) \[41\] was used. However, there are several issues that should be mentioned regarding this procedure. First of all, this is an indirect evaluation, which means that the results of the pseudo-alignment can be distorted by other algorithms in the pipeline, i.e. by the cluster algorithm or by the metric used to compare the two trees. Secondly, the *Robinson-Folds metric* only measures the topological distance between a pair of trees which means that branch lengths are ignored. While the idea that better distances lead to better phylogenies sounds reasonable, contradictory results have been reported recently, where better distances led to worse phylogenies \[35\]. The last problem with this evaluation method is that it is difficult to find reliable reference trees. Since the evolutionary history is generally unknown, the resulting trees should be regarded as an alternative suggestion if they differ from the (probably wrong) reference tree instead to be considered incorrect. This, however, makes it hard to assess the performance of the pseudo-alignment approach to recover phylogenies. Therefore, one artificially generated set of DNA sequences was generated using the software *Alf* \[8\]. *Alf* is a simulation framework that mimic genome evolution. To generate a DNA data set, 1500 genes were mutated along an evolutionary tree, using *Alf*. Genomic events a lateral gene transfer, gene duplications and deletions as well as rate viability between different genes were used to generate a data set which is close to real-world genomes. The advantage is that the phylogeny of the organisms is known, and therefore the true reference tree is available. All the 1500 genes were concatenated to generate one data set with a total size of about 42mb and 30 sequences.

As real-word data sets, three different classes of sequence data were used. One procaryotic data set, consisting of 29 *E.coli/Shigella* genomes with a total length of about 142mb, one eukaryotic data set, consisting of 14 plant genomes of the *brassicales* clade with a total length of about 4.2gb and finally one mitochondrial data set of 27 primates with a total length of about 446kb.

As shown in the introduction, there are now numerous alignment-free methods available but only the newest generation of methods are able to define distances based on an evolutionary model. Since these methods are considered to be the most reliable, the pseudo-alignment is only compared to these methods. This includes the estimator $k_r$ \[20\] and the extended version of it, called *andi* \[19\], the *spaced-words* distance estimator $d_N$ \[35, 36\] and the *CO-Phylog* \[58\] approach.

For the evaluation pipeline the implementations provided by the *PHYLIP* package \[15\] were used. In particular, the program *neighbor* which implements the *neighbor joining algorithm* \[43\] was used as well as the program *treedist* which implements the *Robinson-Folds metric* \[41\]. Additionally, the trees shown were midpoint rooted using the program *retree*.

### 4.2 Results

Figure 4.1 shows the estimated distances by the pseudo-alignment approach for pairs of artificially generated sequences with a known number of substitutions per site. As can be seen, the distances
Figure 4.1: Estimated distances based on the pseudo-alignment approach are shown as a function of the number of substitutions per site \((K)\). The red line shows the correct substitution rate. Are very accurately estimated up to a substitution rate of about 0.5. There are almost no error bars and thus the estimated distances can be considered to be very reliable. However, for larger substitution rates the distances are still increasing but significantly underestimated. The reason for this observation is clear: It is because of the cut off defined by the filter. This cut off was set deliberately to this value because all background matches are filtered out but for higher substitution rates, homologous matches are wrongly filtered out as well. Therefore, the pseudo-alignment only estimates distances up to 0.5 substitutions per site reliably. In this study, all compared sequences had a smaller substitution rate than 0.5 and therefore, the pseudo-alignment should produce meaningful results.
Figure 4.2: Estimated distances based on various alignment-free methods are shown as a function of the number of substitutions per site ($K$). Insertions and deletion were incorporated with a chance of 1% per position and a length between 1–50 nucleotides. The red line shows the correct substitution rate.
4.2. RESULTS

Figures 4.2 and 4.3 show the estimated distances by the pseudo-alignment approach as well as the distances by several other alignment-free methods. In contrast to the previous evaluation, insertions and deletions were incorporated into the sequences. The first observation is that all methods fail to estimate the correct distances reliably. The program andi systematically underestimates the distances for smaller substitution rates while for larger substitution rates no reasonable distances are obtained any more. The spaced-words distance estimator and kr perform better than andi but overestimate the distances clearly. CO-Phylog estimates the distances most accurate up to about 0.12 substitutions but then underestimates the distances. The pseudo-alignment approach also overestimates the distances but not as much as the spaced-words estimator and kr. If data sets with small substitution rates are to be analysed then CO-Phylog might outperform the other programs.
Figure 4.4 shows the results for the phylogeny analysis of the artificial data set generated by \textit{ALF}[8]. In addition to the sequences, \textit{ALF} also provides the reference tree. Therefore, the RF distance between the reference tree and the trees reconstructed by various \textit{alignment-free} methods is shown. The smaller the RF distance is the more closely is the reference tree approximated. As can be seen, \textit{Kr} approximates the reference tree most accurate with a RF distance of only 12. \textit{Andi}, however, which should replace \textit{Kr} performs less accurate then all other \textit{alignment-free} methods. \textit{CO-Phylog} achieves a RF distance of 16 and the \textit{spaced words} distance estimator varies around 15 which is slightly better than the \textit{pseudo}-alignment.

The first real word benchmark data set used in this study consists of mitochondrial genomes of 27 primates. This data set was firstly analysed by [20] and later used as benchmark data set by [28] and [35]. The average length of each sequence is about 16\,kb which is short enough to calculate a \textit{multiple sequence alignment} (MSA) using \textit{Clustal}\textOmega [45]. Based on this MSA, evolutionary distances were estimated using the program \textit{dnadist} from the \textit{PHYLIP} package [15]. Finally, the \textit{NJ} algorithm was applied to the resulting distance matrix to obtain the reference tree. A MSA calculated with \textit{Clustal}\textOmega can be considered to be reliable and therefore, the distances and the resulting tree should be reliable as well. Figure 4.5 shows the RF distance of the reconstructed trees and the reference tree. The approaches \textit{Kr} and \textit{Andi} perform poorly on this data set and also \textit{CO-Phylog} resulted in a RF distance of 4. The winner is clearly the \textit{spaced words} distance estimator because the recovered trees often coincide with the reference tree. However, the error bars show that is very important which
set of patterns is used. A similar result was achieved for the \textit{pseudo-alignment} approach although the \textit{RF} distance is on average higher as for the \textit{spaced words} distance estimator but significantly better than the other \textit{alignment-free} methods.

![Robinson-Foulds Distance](image)

**Figure 4.5:** The Robinson-Foulds distance is shown between the reference tree and trees reconstructed by various \textit{alignment-free} methods for a set of 27 mitochondrial genomes.

The \textit{E.coli/Shigella} data set is used in many alignment-free studies as benchmark dataset \cite{19, 35, 58} because the reference phylogeny of these organisms is very well studied and can be considered to be reliable. To obtain a meaningful topology, the authors of \cite{59} extracted the 2034 core genes, i.e. the genes shared by all organisms, and aligned them to apply a \textit{maximum likelihood} to infer the phylogeny. This topology agrees with the topology based on the genome alignment produced by \textit{mugsy} \cite{4} which is shown in figure 4.6A. Since both slow but accurate methods lead to the same topologies, it can be regarded as reliable reference. Figure 4.6 and figure 4.7 show the different phylogenies obtained from different methods. The estimator $k_r$ was omitted in this analysis because this methods does not consider the reverse complement of the sequences which is important for this data set. As can be seen in the figures, \textit{andi} as well as \textit{CO-Phylog} approximates the reference tree very precise but both methods, however, were not able to restore the clade marked by C. Additionally, \textit{CO-Phylog} also failed to cluster \textit{E.coli UMN026} correctly. The \textit{spaced-words estimator} performed very poorly on this data set which is contradictory the results obtained by the \textit{mtDNA} data set as well as the simulated sequences by \textit{ALF}. When comparing the Robinson-Foulds distances, then \textit{andi} only has a RF distance of 4 as well as \textit{CO-Phylog} while the \textit{spaced-words estimator} yields a distance of 10 compared to the reference tree. The phylogeny based on the \textit{pseudo-alignment}, shown in figure 4.7 leads to a RF distance of 2 and therefore differs only in one branch, marked by
C. While all alignment-free methods clustered the strains E. coli 536 and E. coli CFT073 together, the pseudo-alignment correctly identified E. coli 536 as a more distantly related organism. However, pseudo-alignment failed to cluster E. coli CFT073 and E. coli Ed1a together but the reference tree is more closely approximated than by the other alignment-free methods.

Figure 4.6: Phylogenies of 29 E. coli/Shigella genomes computed by different methods. Figures A-C were obtained from [19]. A is based on a mugsy alignment [9]. B is based on distances computed by CO-Phylog, C is based on distances computed by andi and D is computed based on distances by the spaced-words estimator.
4.2. RESULTS

Figure 4.7: Phylogenies of 29 E.coli/Shigella genomes calculated by distances based on the pseudo-alignment

The final challenge for the pseudo-alignment is a data set consisting of 14 plant genomes. This data set was previously analysed in [17] and used as benchmark data set in the study [28]. The results of this study can be seen in figure 4.8A-F. Figure 4.8G shows the phylogeny of the spaced-words distance estimator and figure 4.8H shows the phylogeny of the pseudo-alignment approach. Because of the huge size of this data set, the programs were run on a different machine on which it was not possible to install andi and Co-Phylog. Therefore, these methods are omitted in the evaluation. However, since both programs are designed to work on very closely related sequences and the plants are the most distantly related data set analysed in this work, these programs might not produce meaningful trees anyway. As reference tree for the plant genomes a phylogeny based on a protein alignment and maximum likelihood was used, shown in 4.8A. No alignment-free method was able to recover the reference tree but significant differences between different alignment-free methods can be observed. The tree recovered by the ACS approach [52], figure 4.8B, clustered 3 of 4 branches correctly and only failed to assign Carica papaya to the Brassicales clade. However, the branch length are very short for most branches and therefore, the quality of this tree is ambiguous. The phylogeny reconstructed by FFP [46], figure 4.8C and, in particular, the tree inferred by \( k_r \), figure 4.8D differ substantially from the reference tree. The tree obtained by the spaced-words approach using the Jensen-Shannon divergence, figures 4.8E-F, seems to be slightly more accurate than FFP and ACS but the branch lengths look suspicious, because they are very short.

If, instead of the Jensen-Shannon divergence, the new evolutionary distance is used for the spaced-words approach the accuracy of the tree worsened, see 4.8G. The tree based on the pseudo-alignment
Figure 4.8: Results of the phylogeny analysis of 14 plant genomes. Trees A-F were obtained from [28]. A: Shows the reference tree based on a protein alignment, created by [17]. The trees B-F show phylogenies of different alignment-free methods, whereby B is the tree recovered by the ACS approach, C FFP, D kr, E spaced-word with the Jensen-Shannon divergence and contiguous-word frequencies and F similar to E but based on spaced-word frequencies. G is based on the spaced-words distance estimator and H is based on the pseudo-alignment approach.

is shown in figure B-H. The topology is similar to the topology obtain by spaced-words using the Jensen-Shannon divergence. However, the branch lengths of the tree based on pseudo-alignment appear to be more precise compared to the reference tree than the tree build by the spaced-words approach. Therefore, one could argue that the tree based on the pseudo-alignment is more accurate compare to the trees obtained by the other alignment-free methods.
Chapter 5

Conclusion and Outlook

Standard methods for sequence analysis and phylogeny reconstruction are based on (multiple) sequence alignments. While these methods are known to be very accurate when applied to a set of homologous genes or proteins, they face notorious difficulties if whole genomes are to be analysed. Because of their slow run time and genomic events as rearrangements, gene duplication/deletion and inversions these methods reach their limits. Therefore, alignment-free methods are increasingly used for whole genome sequence comparison and phylogenetic reconstruction. While these methods are very fast, they are in general less accurate than alignment-based methods. In this work, the weaknesses and strengths of current state of the art alignment-free methods were discussed in chapter 2.

The main part of this thesis was to develop and alignment-like approach, called the pseudo-alignment approach which addresses the problems of alignment-free methods as well as alignment-based methods. Most alignment-free methods compare the word composition of the sequences or consider the length of common subwords to define a distance. Most recent alignment-free methods are in one or the other way based on inexact word matches as the spaced-words approach [28][35][36]. Instead of counting contiguous words frequencies, this approach counts spaced word frequencies according to a predefined pattern, consisting of match and don’t care positions.

Spaced-words are a key component for the pseudo-alignment approach. Spaced-word matches between pairs of sequences are determined and the nucleotides occurring at the don’t care positions lead to the pseudo-alignment of two sequences. Instead of aligning homologous positions to each other as in a real alignment, the pseudo-alignment ignores the positions and aligns homologous nucleotides directly. Therefore, the computationally expensive chaining of positions is avoided and the run time is much lower. To ensure that only homologous nucleotides are aligned to each other, two techniques were developed to distinguish between background matches and homologous matches. While one technique counts how often a nucleotide was found, the other technique analyses the don’t care positions of spaced-word matches to decide if a match occurred due to homology or just by
Finally, the pseudo-alignment approach was evaluated using artificial as well as real-word data sets. Tests on simulated pairs of sequences showed that the expected distances were precisely estimated which suggests that this approach is able to determine evolutionary relationships among species. To show this, the second part of the evaluation was carried out by phylogeny analysis. Tests on simulated as well as real data sets showed that the recovered phylogeny based on the pseudo-alignment is competitive with other alignment-free methods. For whole genomes, the pseudo-alignment approach outperformed most other methods while for one artificial data set as well as a set of mitochondria genomes it performed slightly inferior compared to other methods.

The pseudo-alignment approach is a novel method for alignment-free sequence comparison and phylogeny reconstruction. Tests have shown that it produces good results and can help to analyse and compare large scale data sets. This novel methods can bring insights of evolutionary relationships among a group of species which could interesting for the scientific community.

One way to continue this project is to generalize the pseudo-alignment approach to a set of sequences, i.e. to calculate a multiple pseudo-alignment. One way to do this is to find spaced-word matches which occur in all sequences. However, the drawback is that for more distantly related sequences the chance to find a match in all sequences together is very small. Therefore, only very few matches would be found and consequently the pseudo-alignment would be very short. Another idea could be to align sequences to a reference sequence step by step. That is, all sequences are aligned independently to one reference sequence. Then the columns are extracted and this procedure can be repeated for another reference sequence. Finally, multiple pseudo-alignment can be defined as the set of columns that were found for all reference sequences. However, the run time of this approach would be slightly higher as well as the memory usage but the advantage would be that methods like maximum likelihood can be applied to infer phylogenies.
Bibliography


