Filtered spaced-word matches: a novel approach to fast and accurate sequence comparison

Dissertation
for the award of the degree
„Doctor rerum naturalium“
of the Georg-August-Universität Göttingen
within the doctoral program
„Microbiology and Biochemistry“
of the Georg-August-University School of Science (GAUSS)

submitted by
Chris-Andre Leimeister

from
Ludwigsburg

October 31, 2018

Members of the Thesis Committee

Prof. Dr. Burkhard Morgenstern (1st Referee)
Institute for Microbiology and Genetics, Department of Bioinformatics, Georg-August-Universität Göttingen

Dr. Johannes Söding (2nd Referee)
Quantitative and Computational Biology, Max-Planck Institute for Biophysical Chemistry Göttingen

Prof. Dr. Tim Beißbarth
Department of Medical Bioinformatics, University Medical Center Göttingen

Further Members of the Examination Board

Prof. Dr. Jörg Stülke
Institute for Microbiology and Genetics, Department of General Microbiology, Georg-August-Universität Göttingen

Prof. Dr. Anja Sturm
Institute for Mathematical Stochastics, Georg-August-Universität Göttingen

Prof. Dr. Stefan Klumpp
Institute for Nonlinear Dynamics, Theoretical Biophysics, Georg-August-Universität Göttingen

Date of the oral examination: December 12 th 2018

Affidavit

I hereby confirm that this thesis has been written independently and with no other sources and aids than quoted.

Göttingen, 31. October 2018

Chris-Andre Leimeister

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Doktorarbeit unterstützt und motiviert haben.

Mein Dank gilt zuerst Prof. Dr. Burkhard Morgenstern, meinem Doktorvater, für die Betreuung dieser Arbeit und die andauernde Unterstützung seit meinem Bachelorstudium. Insbesondere der kreative Austausch von Ideen hat mich immer sehr motiviert und wird mir sehr lange positiv in Erinnerung bleiben.
Ich danke insbesondere Dr. Johannes Söding und Prof. Dr. Tim Beißbarth für die konstruktive Kritik und bereichernden Ideen in den Komiteetreffen. Des Weiteren danke ich Dr. Peter Meinicke und Heiner Klingenberg für hilfreiches Feedback bei meinen Präsentationen. Ich bedanke mich bei Prof. Dr. Jörg Stülke für die Möglichkeit im GGNB Programm *Microbiology and Biochemistry* promovieren zu dürfen. Außerdem bedanke ich mich bei meinem Kollegen und Freund Thomas Dencker für die vielen hilfreichen Gespräche, sowie bei Lars Hahn für die hilfreichen Diskussionen.

Mein ganz besonderer Dank gilt meinen Eltern, Inge und Jürgen, die mich über die Jahre hinweg stets unterstützt haben, sowie meinen Geschwistern Marco und Julia. Tief verbunden bin ich meiner Freundin Bingyao, auf die ich mich auch in schwierigen Zeiten immer verlassen konnte und die stets hinter mir stand.

Contents

Publication List

Journal Papers

- **Chris-Andre Leimeister** and Burkhard Morgenstern. *kmacs*: the k-mismatch average common substring approach to alignment-free sequence comparison. *Bioinformatics*, 30:2000–2008, 2014.
- **Chris-Andre Leimeister**, Marcus Boden, Sebastian Horwege, Sebastian Lindner, and Burkhard Morgenstern. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics*, 30:1991–1999, 2014.
- **Chris-Andre Leimeister**, Salma Sohrabi-Jahromi, and Burkhard Morgenstern. Fast and accurate phylogeny reconstruction using filtered spaced-word matches. *Bioinformatics*, 33:971–979, 2017.
- **Chris-Andre Leimeister**, Thomas Dencker, and Burkhard Morgenstern. Accurate multiple alignment of distantly related genome sequences using filtered spaced word matches as anchor points. *Bioinformatics*, 2018, doi: 10.1093/bioinformatics/bty592. in press.
- **Chris-Andre Leimeister**, Jendrik Schellhorn, Svenja Schöbel, Michael Gerth, Christoph Bleidorn and Burkhard Morgenstern. *Prot-SpaM*: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *GigaScience*, 2018. to appear.
- Sebastian Horwege, Sebastian Lindner, Marcus Boden, Klaus Hatje, Martin Kollmar, **Chris-Andre Leimeister**, and Burkhard Morgenstern. *Spaced words* and *kmacs*: fast alignment-free sequence comparison based on inexact word matches. *Nucleic Acids Research*, 42:W7–W11, 2014.
- Burkhard Morgenstern, Bingyao Zhu, Sebastian Horwege, and **Chris-Andre Leimeister**. Estimating evolutionary distances between genomic sequences from spaced-word matches. *Algorithms for Molecular Biology*, 10:5, 2015.

- Lars Hahn, **Chris-Andre Leimeister**, Rachid Ounit, Stefano Lonardi, and Burkhard Morgenstern. *rasbhari*: optimizing spaced seeds for database searching, read mapping and alignment-free sequence comparison. *PLOS Computational Biology*, 12(10): 1-18, 2016.

- Burkhard Morgenstern, Svenja Schöbel, and **Chris-Andre Leimeister**. Phylogeny reconstruction based on the length distribution of k-mismatch common substrings. *Algorithms for Molecular Biology*, 12(1):27, 2017.

Conference Papers (peer reviewed)

- **Chris-Andre Leimeister**, Marcus Boden, Sebastian Lindner,Sebastian Horwege and Burkhard Morgenstern Fast alignment-free sequence comparison using spaced-word frequencies. In *German Conference on Bioinformatics 2015*, highlight talk.

- **Chris-Andre Leimeister**, Salma Sohrabi-Jahromi and Burkhard Morgenstern Fast and Accurate Phylogeny Reconstruction using Filtered Spaced-Word Matches. In *German Conference on Bioinformatics 2017*, highlight talk.

- Marcus Boden, Martin Schöneich, Sebastian Horwege, Sebastian Lindner, **Chris-Andre Leimeister**, and Burkhard Morgenstern. Alignment-free sequence comparison with spaced k-mers. In *German Conference on Bioinformatics 2013*, volume 34 of *OpenAccess Series in Informatics (OASIcs)*, pages 24–34, 2013.

- Burkhard Morgenstern, Bingyao Zhu, Sebastian Horwege, and **Chris-Andre Leimeister**. Estimating evolutionary distances from spaced-word matches. In *Algorithms in Bioinformatics*, volume 8701 of *Lecture Notes in Computer Science*, pages 161–173. Springer, 2014.

- Thomas Dencker, **Chris-Andre Leimeister**, Michael Gerth, Christoph Bleidorn, Sagi Snir, and Burkhard Morgenstern. Multi-SpaM: a Maximum-Likelihood approach to phylogeny reconstruction using multiple spaced-word matches and quartet trees. In *Comparative Genomics*, pages 227–241. Springer International Publishing, 2018.

Abstract

Standard methods for biological sequence comparison and phylogeny reconstruction are traditionally based on sequence alignments. These methods are very accurate but also computationally expensive. Because of the exponentially growing amount of biological sequence data, alignment-free methods have become more important over the past decades. Alignment-free methods are substantially faster than alignment-based methods and are essential for large scale sequence comparison. One major application of alignment-free methods is whole genome phylogeny reconstruction. To this end, distances between pairs of genomes are calculated and subsequently clustered. Current alignment-free methods are fast but less accurate than alignment-based approaches.

In this thesis, I developed the *filtered spaced-word matches (FSWM)* approach, a new alignment-free method for fast and accurate whole genome phylogeny reconstruction. *FSWM* rapidly identifies spaced-word matches which are defined by patterns of *match* and *don't care* positions. The fraction of non-matching nucleotides at the *don't care* positions are used to estimate evolutionary distances. To reduce the noise from random matches, I developed a filtering technique which calculates a similarity score for each spaced-word match and discards matches with a score below a threshold. This filtering removes most of the unwanted background matches and the distances calculated based on the remaining spaced-word matches are very accurate.

Moreover, I investigated if *FSWM* can be used to identify anchor points for genome alignments. I integrated a slightly modified version of *FSWM* into *mugsy* [6], a popular multiple-genome-alignment pipeline. If *FSWM* is used to identify anchor points, more homologies are found and aligned and the alignments are of higher quality.

Furthermore, I transferred the idea of *FSWM* from genomic sequences to protein sequences. I developed *Prot-SpaM*, a fast tool which estimates evolutionary distances between pairs of whole proteoms. *Prot-SpaM* is the first alignment-free tool that estimates the number of substitutions between pairs of protein sequences without sequence alignment.

1 Introduction

The comparison of different organisms has always been a fundamental task in biology. In many popular scientific writings one can read that the observation and comparison of different beak forms and functions of finches led to Charles Darwin great work „On the Origin of Species “ [18]. Even though it is controversial if the comparison of the finches was responsible for his breakthrough [112], it symbolizes how important comparative studies always have been for the study of life.

Nowadays, organisms are usually compared on a molecular level. Instead of similarities or differences in the morphological or anatomical characteristics, differences in the DNA, the hereditary material, or proteins are used to determine the degree of relatedness. Zuckerkandl and Pauling [137] were among the first who discussed that information about the evolutionary history can be gained if homologous residues of DNA or protein sequences are compared. Today, thanks to the convergence of two major technological revolutions, next generation sequencing and high performance computing, sequence comparison has become cheaper and faster than ever before [9]. Because of these developments the major objective to reconstruct the evolutionary history of all life on earth has taken a major step forward. Evolutionary relationships among different organisms are depicted as phylogenetic trees or networks. Since it is assumed that all living organisms on earth share one last universal common ancestor [119] the goal is to unite all lineages into one single phylogenetic tree or network which is called the tree of life [41].

Homologous residues between a set of sequences are traditionally identified by sequence alignments. There are, in general, two ways to reconstruct the phylogeny based on a sequence alignment: distance-based methods which require a subsequent clustering of the sequences and character-based methods which infer the phylogeny directly from the alignment. The distance-based methods often estimate the number of substitutions between pairs of sequences to quantify their level of relatedness. If a sequence alignment is given, the nucleotides, which changed into another one, are revealed and the mismatch rate between pairs of sequences can be calculated easily. This observed rate, however, does not reflect the correct number of substitutions that have happened since the organisms diverged. That is because nucleotides can change back to their original base. For example, a nucleotide can change from A to T and then back to A again. In this scenario, zero

mutations are visible but in reality two mutations have happened. To solve this problem, substitution models were developed to turn the observed number of mismatches into an evolutionary distance which reflect the number of substitutions. The first substitution model was proposed by *Jukes&Cantor* [48]. They assumed that there are equal mutation rates among the four nucleotides. In 1980, another substitution model was proposed by *Kimura* [53] which distinguishes between transitions and transversions. Since then multiple other substitution models have been proposed [116, 25, 34], addressing, for example, unequal expected nucleotides frequencies.
After all pairwise distances are calculated a tree reconstruction algorithm must be used to recover the phylogeny. Most commonly used is the *Neighbor-Joining (NJ)* algorithm [97]. One advantage of this algorithm is that is does not assume a molecular clock compared to other tree reconstruction algorithms such as *UPGMA* [106].
Character-based methods which infer the phylogeny directly from a multiple sequence alignment are, for example, *Maximum Likelihood* [108], *Maximum Parsimony* [28] or *Bayesian Inference* [95]. The idea of the *Maximum Parsimony* approach is to identify the tree which implies the smallest number of substitutions. *Maximum Likelihood* methods works broadly similar to *Maximum Parsimony* but instead of the number of substitutions as criterion, different substitution models can be used. *Bayesian inference* methods use *Markov chain Monte Carlo sampling* which is based on a prior probability distribution of the possible trees. These methods are considered to be more accurate than the distance-based methods but are also considerably slower.

Alignment-based phylogeny is well studied [26] and still considered as the gold standard. There are, however, notorious difficulties associated with sequence alignments. Zielezinski et al. [135] identified five cases where sequence alignments reach their limits: first, sequences must be homologous to be aligned. Therefore, homologous regions must be identified first which require manual intervention and often prior knowledge of the sequences under study. Second, the authors state that sequence alignments become increasingly imprecise the more divergent the sequences are. Third, alignment-based methods are generally computationally expensive. The time complexity of an optimal pairwise alignment is proportional to the product of the sequence length. Given the huge volume of sequence data generated by modern sequencing technologies, an excessive amount of computational resources would be required to align all the sequences. Fourth, the calculation of an optimal multiple sequence alignment is NP-hard [124]. Therefore, heuristics are needed which do not ensure optimal-

ity of the alignment and this effects further downstream analyses. The fifth problem they identified is that multiple a priori assumptions are needed to tune the parameters such as gap penalty or the choice of the substitution matrix.
An alternative to alignment-based methods are alignment-free methods. Zielezinski et al. [135] defined alignment-free methods as follows: *'(...) any method of quantifying sequence similarity/dissimilarity that does not use or produce alignment (assignment of residue-residue correspondence) at any step of algorithm application'*. This definition covers a broad spectrum of different fields in bioinformatics where alignment-free methods are used. In this work, however, the term alignment-free is always used in the context of phylogeny reconstruction. Despite the quoted definition above, there are alignment-free methods for phylogeny reconstruction that explicitly establish a residue-residue correspondence to define distances between pairs of sequences. These methods are discussed and described in detail in Section 1.1.4.
The biggest advantage of alignment-free methods over alignment-based methods is their lower run time. Therefore, they are often used for large scale sequence comparison where alignment-based methods reach their computational limits. One major application of alignment-free methods is whole genome phylogeny reconstruction. Alignment-free methods can be directly applied to whole genomes because they are not sensitive to rearrangements and gene duplications. Moreover, no prior knowledge about orthologies or the level of relatedness is required which reduces preprocessing of the data to a minimum. However, alignment-free methods are generally less accurate than alignment-based methods. Therefore, Haubold et al. [37] stated that alignment-free methods trade speed for precision.

In my thesis, I focus on the development of new a alignment-free approach to bridge the gap between speed and accuracy of alignment-based and alignment-free approaches. For a better understanding, I start with a comprehensive review of alignment-free methods, beginning with the first very basic ideas and ending with the latest state-of-the-art approaches. Some recently developed, so-called, alignment-free methods are in a twilight zone: they calculate distances based on *short local gap-free alignments* but still considered to be alignment-free. They strive for the accuracy of alignment-based methods but try to circumvent their disadvantages. I describe these methods in Section 1.1.4 in detail. The introduction is followed by the main work of my thesis, the development of the *filtered spaced-word (FSWM)* approach. In Chapter 3, I describe how *FSWM* can be applied to whole genome alignments as anchor points and in Chapter 4 I extended the *FSWM* ap-

proach to protein sequences. My thesis ends with a discussion and an outlook where I give suggestions how this approach can be continued in the future.

1.1 Alignment-free methods for phylogeny reconstruction

In the following sections, I give an overview of well-known approaches to alignment-free sequence comparison. I start by describing the traditional classification of these methods according to several review papers. This classification divide alignment-free methods into word count methods and match lengths methods. Methods based on word counts are described in detail in Section 1.1.2 and match lengths methods in Section 1.1.3. Recently, new approaches emerged which estimate distances based on so-called *micro-alignments*. These new methods are described in Section 1.1.4. In this section, I also explain the fundamental differences between methods based on micro-alignments and methods based on word counts or match lengths. Moreover, I discuss their advantages over traditional alignment-free methods but I also point out their shortcomings.

1.1.1 Classification

The first comprehensive review of alignment-free methods was published in 2003 [122]. These authors identified two main categories of alignment-free methods: methods based on (relative) word frequencies and methods that are not dependent on a specific resolution. Word frequency methods count words of a fixed length for each sequence and define distances between the word frequency vectors as the distances between the corresponding sequences. The other class of methods are called resolution free because no parameter for the word length must be specified. These methods define distances based on the amount of information shared between a pair of sequences. This quantity can be approximated by joint compression of the sequences. Details of this procedure are described in Section 1.1.3. The theoretical background of data compression is based on information theory [100]. Originally, information theory was developed to study the transmission of messages over a noisy channel but nowadays it is used in many different fields. In bioinformatics, information theory is not only used for sequence comparison but also to address other issues such as, for instance, prediction of transcription factor binding sites [114]. A review of impor-

tant applications of information theory for biological sequence analysis was published in 2014 [121].

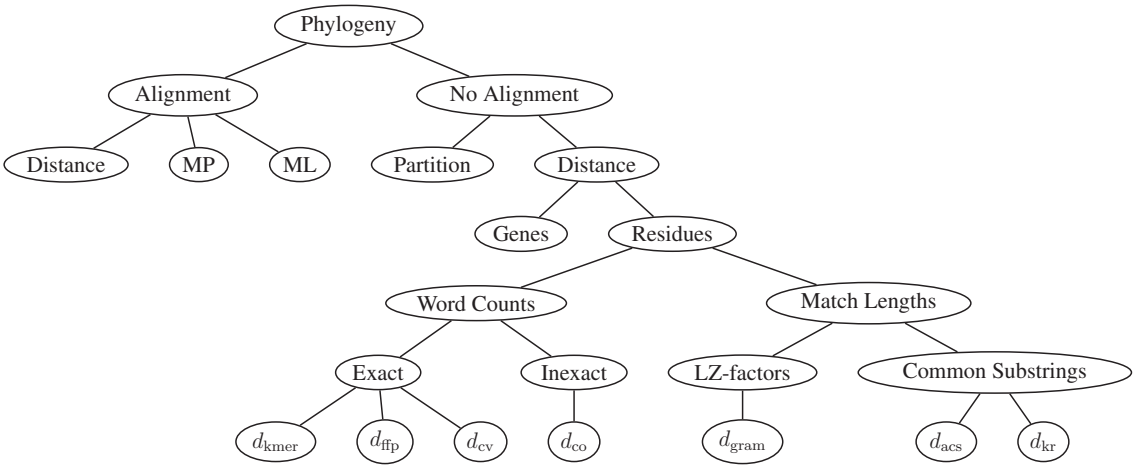

Figure 1: Classification of phylogeny reconstruction methods, taken from Haubold [35]. *MP: maximum parsimony*, *ML: maximum likelihood*, *LZ: Lempel-Ziv* [136], d_{kmer}: k-mer based approach [132], d_{ffp}: *feature frequency profile* [103], d_{cv}: *CVTree* [138], d_{CO}: *CO-Phylog* [133], d_{gram}: grammar-based [96], d_{ACS}: *Average common substring approach* [120], d_{K_r}: K_r [37]

About one decade later, a new review of alignment-free methods was published [35]. In this review, a general classification of phylogeny reconstruction methods was proposed, including alignment-based and alignment-free approaches. This classification is shown in Figure 1. It illustrates various different approaches to phylogenetic reconstruction but the main focus is on residue-based alignment-free methods which they divided into two categories: word counts (frequencies) and match lengths. In this classification, the match lengths category can be seen as a superclass of the compression-based methods and, in fact, the author classified *LZ-factorisation* [136], a compression algorithm, as a match lengths method. Moreover, match lengths methods are resolution free and therefore one can argue that the proposed classification agrees with the classification of the first review by Vinga and Almeida [122].

The alignment-based methods shown in Figure 1 were briefly reviewed in the introduction. These methods are not the focus of this work and omitted in the following sections.

Almost all alignment-free methods calculate pairwise distances from which the tree is inferred. There is one exception which is denoted as *partition-based* [42] in Figure 1. Un-

til very recently, the *partition-based* approach was the only alignment-free method that inferred trees without computing pairwise distances first. While writing this thesis, another alignment-free non–distance-based method was presented at the *RECOMB-GC 2018*, called *Multi-SpaM* [22]. It rapidly identifies homologous blocks between four sequences each and calculates quartet trees using *RAxML* [108]. Then these quartet trees are amalgamated into one tree using the *Quartet MaxCut* algorithm [104]. One drawback is that *Multi-SpaM* is only able to recover the topology but not the branch lengths.
The alignment-free distance-based methods in Figure 1 are divided into gene-based and residue-based methods. Most common gene-based methods quantify the number of shared or absent genes among the genomes [118] or are based on comparative gene mapping [83]. The problem with these methods is that the genes must be identified first which either require gene annotations or they must be found by sequence alignments. This is contradictory to the idea of alignment-free methods and therefore, the author of Figure 1 states that it would be *'a slight overstatement to call them alignment-free'*. Consequently, most work has been done on residue-based methods. These methods are described and discussed in detail in the following three sections.

Another review paper about alignment-free methods was published in 2017 [135]. It contains over 180 references which shows that alignment-free sequence comparison is currently a hot topic and new methods are developed rapidly. This review is more general than the other reviews and covers multiple different domains where alignment-free approaches are used. This includes for example read alignment [2, 58, 70], protein classification [14, 64, 72, 73], isoform quantification from RNAseq reads [89], sequence assembly [134], metagenomics [5, 13, 66, 78, 115, 117, 125, 131], analysis of regulatory elements [24, 50, 65] and identification of biomarkers [23]. In my work, however, I concentrate on alignment-free methods for phylogeny reconstruction.
Other reviews focused on statistical analyses of popular alignment-free distance measures [10, 99], assessed the performance of alignment-free methods in the presence of lateral gene transfer [7] or explored alignment-free methods for short unassembled reads [107]. The latest review of alignment-free approaches was published in 2018 [94] and concentrated on word count methods for next generation sequencing data.

The vast majority of alignment-free methods uses either word counts or match lengths

Word Counts	Match Lengths	Micro-Alignments
FFP [103]	ACS [120]	CO-Phylog [133]
CVTree [138]	grammar [96, 87]	andi [38]
Spaced [44, 60, 81]	K_r [37]	FSWM [61]
D_2^* [74, 93, 123]	*kmacs* [59, 82]	Prot-SpaM [63]
	UA [15]	Multi-SpaM [22]

Table 1: Alignment-free methods based on different information sources.

as basis for the distance calculation. There is one remarkable exception, called *CO-phylog* [133]. This approach can be found in Figure 1 as word count method under the name d_{CO}. *CO-phylog* estimate distances from *micro-alignments* which are short local gap-free alignments. The underlying information source from which the distances are calculated is different compared to word count or match length methods. Therefore, I divided alignment-free methods in three different categories: word count methods, match length methods and methods based on micro-alignments, see Table 1. In this table, there are four more approaches, besides *CO-phylog*, the pioneering approach, that are based on micro-alignments: *andi* [38], *FSWM* [61], *Prot-Spam* [63] and *Multi-Spam* [22]. *CO-phylog* and *andi* are described in Section 1.1.4. In this section, I also discuss the similarities and difference of methods based on micro-alignment as well as their advantages and disadvantages. Moreover, I point out the limitations of *CO-phylog* and *andi* which motivates the development of *filtered-spaced word matches (FSWM)* a new alignment-free method, the main objectiv of this thesis (see Section 2). *Prot-Spam* is a modification of the *FSWM* approach and described in section 4. *Multi-Spam* generalizes the *FSWM* approach to multiple sequences and is not a part of this thesis.

1.1.2 Methods based on word counts

Word count methods define distances based on the word composition of the sequences under study. That is, for a fixed word length k the (relative) frequencies of all words of length k are determined. Each sequence is then represented by its frequency vector and distances between these frequency vectors are defined to be the distance between the corresponding sequences. In the literature, words of length k are also often called k-mers, k-tuples or k-grams.

Below is an example for the sequence AATCGAATC and $k = 3$. At first the word frequencies are determined by moving a sliding window of length $k = 3$ from left to right:

```
A A T C G A A T C
A A T
  A T C
    T C G
      C G A
        G A A
          A A T
            A T C
```

During this process all occurring words are counted and stored in a list:

word	AAT	ATC	CGA	GAA	TCG
count	2	2	1	1	1

This process is repeated for all sequences and distances between the lists (frequency vectors) are calculated. Often standard metrics as the *Euclidean distance* or the *Jensen-Shannon divergence* [71] are used. The basic idea is that the closer related two sequences are, the more similar the word frequency distributions and the smaller the distance between the corresponding frequency vectors. Intuitively, this seems to make sense but, on the other hand, it is also clear that these distances do not reflect the number of substitutions that have happened since two organisms diverged, i.e. these distances are not based on a stochastic model of evolution. As pointed out in the introduction, evolutionary relationships are usually defined based on the number of substitutions. Empirically, however, these distances are a good estimate of the evolutionary relationship between organisms. The exploration, why that is the case, is a major part of this section. To this end, I start with a brief historical overview of word count methods which is, however, by no means exhaustive.

The foundation for the first word count method was laid by the creation of so called 16S rRNA oligonucleotide catalogs [105]. In the early beginnings of molecular phylogenetics, distances were calculated based on these catalogs and dendrograms were reconstructed based on the obtained distances [29]. In 1987, Woese [128] showed that there is a (small)

correlation between the distances calculated based on the 16S rRNA oligonucleotide catalogs and distances calculated by sequence alignments. A detailed review of the methods used at that time is given in [92]. In the beginnings, it was common to quantify differences between di- and trinucleotide frequencies, i.e. the word lengths were $k = 2$ and $k = 3$. Popular dis/similarity measures include the χ^2 statistics [8], relative dinucleotide abundance [52, 51], and the D_2 statistics [16]. The idea to use trimers might be based on that fact that triplets of nucleotides encode for one amino acid. If coding regions are to be compared, similarity in the codon frequencies should reflect the level of relatedness. Nowadays, however, alignment-free methods are often applied to whole genomes, including coding and non coding regions. For these sequences, longer word lengths seem to work better. The question of how to find the optimal word length was investigated in [103, 49, 102, 130]. These authors empirically determined a lower and an upper limit of word lengths such that word lengths within this range produce the most promising results. Their tool, called *Feature Frequency Profiles (FFP)*, is one of the most popular implementation of generic word count methods. By default, *FFP* calculates distances with the *Jensen-Shannon divergence* [71] between the frequency vectors but it also provides a wide range of other standard metrics such as the *Euclidean distance*. Recently, a powerful standalone platform for alignment-free sequence comparison was proposed, called *aCcelerated Alignment-FrEe sequence analysis (CAFE)* [75]. It provides a user-friendly graphical user interface and also includes programs to calculate and visualize phylogenetic trees. In *CAFE*, 28 different word count distances are implemented. Some of them take background word frequencies into account. For example, one popular method that adjust their distances for background frequencies is the *Composition Vector Tree (CVTree)* approach [90]. They used a Markov model to predict word frequencies for unrelated sequences and subtract the predicted background frequencies from the observed word frequencies. The idea behind it was to distinguish between random mutations and selective mutations, i.e. mutations which lead to evolutionary advantages. *CVTree* was used in multiple studies, e.g. [91, 33, 30, 69, 113, 139] and is available through a user-friendly webserver [138].

So far, all mentioned approaches are ad-hoc dissimilarity measures, i.e. they do not reflect the number of substitutions between two sequences. To understand the relationship between these rough dissimilarity measures and the substitution rate, it is helpful to take a closer look at the D_2 statistic first.
In 2002, Lippert et al. [74] defined the D_2 statistic to be the number of word matches

of length k between two sequences. This statistic can be easily calculated as the inner product of two word frequency vectors. They showed how the expected number of word matches between two sequences with independent and identically distributed characters can be calculated under the null hypothesis, i.e. between two unrelated sequences. To this end, let L_1 be the length of sequence S_1 and L_2 the length of sequence S_2. Moreover, let q be the random match probability and k be the word length. Then clearly, the expected number of word matches between two unrelated sequences can be calculated by $(L_1 - k + 1) * (L_2 - k + 1) * q^k$. In words: for each position in the first sequence there is a chance for a match at any position in the other sequence. Distance measures that adjust for background hits, such as the popular D_2^* score [93, 123], usually outperform distances which do not take background matches into account. The reason for this is simple: if the background noise is subtracted then the number of the remaining word matches indicate some sort of sequence similarity. Based on this idea, we showed how the expected number of word matches between two *related* sequences can be calculated [81, 80]. To this end, we used a simplified model of evolution with a constant mutation rate p and equal mutation rates of the four nucleotides. Moreover, the sequences are indel-free, have the same length and share homologies over the entire sequence. Under this model, the number of word matches between two sequences, denoted by N, is composed of the number of homologous matches plus the number of random matches. The expected number of word matches can be calculated by

$$\mathbb{E}(N) = \underbrace{(L - k + 1) * (L - k) * q^k}_{background} + \underbrace{(L - k + 1) * (1 - p)^k}_{homolog} \tag{1}$$

This equation shows the relationship between the number of word matches and the mismatch rate p under the simplified model of evolution. Clearly, the mismatch rate $\hat{p}$ can be estimated based on the number of word matches between two sequences with a momentum based approach. By using the *Jukes&Cantor* model, the estimated mismatch rate $\hat{p}$ can be turned into an evolutionary distance $\hat{d}$ that reflects the number of substitution. Therefore, a sequence alignment is not required to calculate evolutionary distances between pairs of sequences. The first approach which was able to estimate evolutionary distances from unaligned genomes was published in 2009, called K_r [37]. Instead of the number of word matches, they estimated substitution rates based on average match lengths. This approach

is described in Section 1.1.3.
Based on Equation 1, it is obvious that the number of background matches grow quadratic with the sequence length while the number of homologous matches only grow linear. There are two more variables that determine whether the number of word matches is dominated by background matches or by homologous matches: the word length k and the probabilities for homologous and background matches. If the match probability for homologous matches is much higher than for background matches, larger word lengths lead to less background hits, hence, reduce the noise. The number of homologous matches, on the other hand, is also decreasing if longer word lengths are used. Therefore, the choice of the word length k is a trade-off between signal and noise. In our previous work [81, 80], we pointed out that if k is *sufficiently* large and the sequences are not too distantly related then the *Jensen-Shannon* divergence approximates $L-N$ and the *euclidean* distance $\sqrt{L-N}$. This answers the question of why these distances lead to evolutionary meaningful distances.

So far, only *contiguous* or *exact* word counts were considered. Inexact word matches, however, play an important role in sequence analysis. Li et al. [67] [68] showed that so-called *spaced-seeds* or *spaced-words* are far superior compared to *contiguous-words* in homology detection. Spaced-words are defined by binary patterns of *match* and *don't care* positions. A spaced-word match occurs between two sequences if all *match* positions coincide. Mismatches are allowed at the *don't care* positions. The advantage of spaced-words over contiguous-words is that they are statistically less dependent on each other and this leads to a higher probability to find a match between two sequences. Later, we used this concept for alignment-free sequence comparison [60, 44]. We proposed to count spaced-word frequencies and defined distances based on these frequency vectors as a measure of global sequence similarity. An example for the sequence AATCGAATC and the pattern 1101 is given below:

A	A	T	C	G	A	A	T	C
A	A	*	C					
	A	T	*	G				
		T	C	*	A			
			C	G	*	A		
				G	A	*	T	
					A	A	*	C

The corresponding spaced-word frequency count is given below:

spaced-word	AA*C	AT*G	CG*A	GA*T	TC*A
count	2	1	1	1	1

The advantage of spaced-word frequencies for alignment-free sequence comparison is that the variance of the number of word matches is smaller compared to contiguous words [80]. Consequently, the estimated distances are more stable if spaced-words are used. We also showed that if multiple patterns are used, the variance can be reduced further. Moreover, the variance is dependent on the patterns, i.e. there are patterns which lead to a smaller variance than others. Therefore, we developed a hill climbing algorithm, called *rasbhari* [32], which tries to minimize the variance of a set of patterns. We showed that the variance of the number of word matches is closely related to a concept called overlap complexity [46, 47].

Word count methods are easy to implement and run very fast. There are two practical solutions to compute word frequencies: One is based on a (rolling) hash function and a data structure to store the word frequencies. The other solution is to sort all occurring words lexicographically so that equal words end up adjacent to each other which can then be counted easily. Once the word counts are determined, it is straight forward to calculate distances based on the frequencies. This simplicity also contributed to the popularity of word count methods.

1.1.3 Methods based on match lengths

In Figure 1, match lengths methods are divided into LZ-factors and common substrings. LZ-factors stands for *Lempel-Ziv factorisation* and is well-known for data compression [136]. Data compression is also used to estimate the relatedness between two sequences. To this end, two sequences are concatenated and then compressed. The level of compression indicates how similar the sequences are. The *Lempel-Ziv factorization* decomposes a sequence into its constituents. To explain this process, some notation is required: For a sequence S, $S[i]$ denotes the i-th element of S and $S[i..j]$ with $i \leq j$ denotes the substring of S, starting at i and ending at j. Then, the *Lempel-Ziv* factors can be determined as follows: If there is a longest match $S[i..j]$ that starts somewhere in $S[1..i-1]$, then the LZ factor

is S[i..j+1] and the process continues at $S[j+2]$. If not, then the factor is $S[i]$ and the process continues at S[i+1]. An example for the sequence S_1 = AATCGAATC is given below:

	A	A	T	C	G	A	A	T	C
Factor 1	A								
Factor 2		A	T						
Factor 3				C					
Factor 4					G				
Factor 5						A	A	T	C

In this example, the sequence S_1 is decomposed into 5 LZ factors (A,AT,C,G,AATC) which is denoted as $c(S_1) = 5$. If an identical sequence S_2 is attached to S_1 then $c(S_1S_2) = 6$, i.e. there is only one additional LZ factor more. This measure, however, increases the more divergent the sequences are. Two popular distance measures based on this concept are [87] and [96].

The other class of match lengths methods define distances between sequences based on the average lengths of longest common substrings. That is, for two sequences S_1 and S_2, these approaches determine for each position i in S_1 the length of the longest common substring, starting at position i in S_1 that matches a substring starting at some position j in S_2. An example is given below:

S_1				A	A	T	**C**	**G**	**A**	**A**	**T**	**C**	**G**	G	C	T	
S_2	G	G	T	A	A	**C**	**G**	**A**	**A**	**T**	**C**	**G**	C	C	G	T	

This examples shows the longest common substring that starts at position 4 in S_1 and matches the substring starting at position 6 in S_2. The length of this substring match is 7. The average of these lengths are a similarity measure of the sequences. The *average common substring approach ACS* [120] turns this similarity measure into a distance based on principles of information theory. In general, these distances are not symmetrical due to difference in sequence length, gene duplications and other genomic events. Therefore, they defined the distance between two sequences as the average of both asymmetric distances. The *ACS* approach was published in 2006 and outperformed most word count methods at

the time. The reason why match lengths methods work so well is that the average substring lengths increases with sequence similarity. If the simplified model of evolution from the previous section (iid, no indels) is used then substrings between two sequences match until a mutation occurs then the exact match ends. It is obvious that if more mutations happen, the lengths of exact matches must be shorter on average than if fewer mutations occur. Haubold et al. [37] used this observation to derive an estimator, called K_r. This estimator is based on the average lengths of *SHortest Unique substring (shustrings)*. Shustrings can be determined by extending the longest common substrings by one position [36]. At first, they derived the probability density function of shustring lengths for two unrelated sequences and then extended it to related sequences. They showed that this estimator estimates substitution rates very precise up to 0.5 substitutions per site. For more divergent sequences, however, this estimator reaches its limits. There are two reasons for it: first, the chance of background matches increases, i.e. the length of shustrings is dominated by background matches instead of homologous matches and second the length of homologous matches and background matches do not differ very much anymore. Therefore, at about 0.5 substitutions per site homologous match lengths become indistinguishable from background match lengths so that distances can not be estimated anymore.
This was the first alignment-free approach ever that was able to estimate the number of substitutions per site. Until this time, all alignment-free distances were ad-hoc measures of dissimilarities and it was believed that a sequence alignment is necessary to infer the number of substitutions between sequences. However, one should not forget that this estimator was derived based on a simplified model of evolution and real-word genomes are far more complex. Not only indels can affect the shustring lengths but also large repetitive regions which are especially frequent in eukaryotic genomes. This issue was addressed by the *underlying subwords approach* [15] by eliminating overlapping substrings. The distances they calculate, however, do not reflect the number of substitutions.

In the previous section, the concept of *inexact matches* was introduced and we developed a similar idea for match lengths methods [59]. Instead of exact longest common substring matches we proposed to search for substring matches with up to k mismatches. This is a generalization of the *ACS* approach, and we called it the *k-mismatch average common substring approach (kmacs)*. There is, however, an inherent problem with this idea: calculating longest common substrings with up to k mismatches can not be done in linear time. Therefore, we suggested to approximate them with a simple heuristic which works

as follows: At first, we search for exact longest common substrings similar to *ACS*. Then, these longest common substring matches are extended without gaps until k mismatches occurred or the end of the sequence is reached. An example for a longest common substring match with $k = 2$ mismatches is shown below:

S_1	C	A	T	**T**	**G**	**C**	**A**	**T**	**A**	**C**	G	A	
S_2			A	**T**	**G**	**G**	**A**	**T**	**C**	**C**	C	G	A	A

In this example, *kmacs* identifies the longest common substring for position 4 in S_1 that exactly matches the substring at some position in S_2. In this case, such a match is found at position 2 in S_2. Then, this match is extended to the right until the third mismatch is reached. The length of this longest common substring with 2-mismatches is 7 but we only count the number of matching nucleotides which is 5 in this case. Then, we averaged all these lengths and defined distances analogous to the *ACS* approach.
A few years later, we derived the length distribution of k-mismatch longest common substrings and the length distribution for the heuristic used in *kmacs* [82]. We developed a tool that estimates the number of substitutions per site based on lengths of the extensions. We showed that this estimator works up to almost 1 substitution per site and therefore, performs better on divergent sequences than K_r.

Like word counts, substring matches can be calculated in linear time. However, more sophisticated data structures are needed such as *suffix trees* [127]. In a *suffix tree* each path from the root to a leaf represents one suffix of a sequence S. To ensure that each suffix is represented by a leaf a special character, usually \$, is appended to the sequence. Otherwise, a suffix that is a prefix of another suffix would not end up in a distinct leaf. *Suffix trees* are one of the most important data structures for pattern matching. They allow extremely fast and important string operations such as string searching. Therefore, they also play an important role in bioinformatics for sequence analysis. Beside alignment-free sequence comparison there are many other applications that rely on *suffix trees*. One example is the rapid identification of *tandem repeats* [31]. Another important application is whole genome alignment. Here, *suffix trees* are used to rapidly find *anchor points* from which the alignment is extended. Perhaps the most popular tool for this is the *MUMmer* pipeline [56]. However, one major disadvantage of *suffix trees* is that they have a relatively high memory requirement of 20 bytes per input character [55]. This problem can be

circumvented if a *suffix array* [76] is used instead. A *suffix array* contains the starting positions of all suffixes of a sequence in lexicographical order. For a constant alphabet, a *suffix array* can be build in linear time with $\mathcal{O}(1)$ workspace [85]. If the *suffix array* is *enhanced* by additional tables, then each string problem that can be solved with *suffix trees* can also be solved with *suffix arrays* with same time complexity [1]. Because *suffix arrays* reduce the memory bottleneck they become more frequently used in bioinformatics [101]. For example, the latest release of *MUMmer* [77] switched from a *suffix tree* implementation to *suffix arrays*. Also most alignment-free approaches described in the section use *enhanced suffix arrays* to identify longest common substrings efficiently.

1.1.4 Methods based on micro-alignments

The term *micro-alignment* was coined by the authors of *CO-phylog* [133]. These methods define distances based on short local gap-free alignments which are identified by spaced-word matches. In this section, two approaches are described: *CO-phylog* and *andi* [38]. The latter one can not be found in Figure 1 because it was published after the review paper. *CO-phylog*, on the other hand, was classified as an *inexact* word count method. It is, to some extent, correct that *CO-phylog* count word matches but only those with a certain property. This procedure is described in the following.

Since *CO-phylog* is based on spaced-word matches, a pattern of *match* and *don't care* positions must be defined first. The patterns they use have a certain structure which is as follows: the patterns start with n consecutive match positions, followed by one single *don't care* position and then again n consecutive match positions. In other words: there is one *don't care* position in the middle of the pattern, flanked by n *match* positions. For example, if $n = 4$ the pattern looks like 111101111. They call the nucleotides at the *match* positions *context* (C) and the nucleotide at the *don't care* position the *object* (O), hence the name *CO-phylog*. An example for a word match between two sequence S_1 and S_2 with the pattern $P = 111101111$ is given below

S_1	A	T	**G**	**A**	**C**	**A**	**T**	**A**	**T**	**C**	**C**	T	A
S_2		C	**G**	**A**	**C**	**A**	**G**	**A**	**T**	**C**	**C**	C	
P			1	1	1	1	0	1	1	1	1		

In the example above, GACA*ATCC is the *context*, colored blue, and T in S_1, respectively G in S_2, is the *object*, colored red. In this case, there is a mismatch at the *don't care* position, i.e. the *object* is different. In general, the nucleotides at the *don't care* position can also match. Therefore, the fraction of spaced-word matches with different objects compared to the total number of spaced-word matches is an estimate of the number of mismatches between the sequences. And this is exactly how the *CO-phylog* distance is defined. For example, if there are 5000 spaced-word matches between the sequences of which 500 have a mismatch at the *don't care* position, i.e. the object differs, and the other 4500 have a match at the *don't care* position, i.e. the same object, then the distances is calculated as $\frac{500}{5000} = 0.1$. Since this distance estimates the p-distances, the authors could have turned it into an evolutionary distance by using the *Jukes&Cantor* model, but they did not do that. The reason might be that this approach was designed for closely related genomes and for small substitution rates, the mismatch rate nearly coincide with the number of substitutions.

There remains one issue with this approach that needs to be addressed: what if there are multiple spaced-word matches with the same *context* but different *objects*? For example lets consider the sequence S_1 = AATGCTGTCCATCCTC and S_2 = AGATGCTTATA and the pattern 11011

S_1	A	**A**	**T**	**G**	**C**	**T**	G	T	C	**A**	**T**	**C**	**C**	**T**	C
S_2			A	G	**A**	**T**	**G**	**C**	**T**	T	A	T	A		

The word AT*CT occurs two times in S_1 with different *objects* and match one word in S_2. In such a case, it is not possible to determine whether a mutation has happened or not because it remains unknown which word match is due to homology and which occurred just by chance (or due to genomic events). Therefore, the authors defined that the word AT*CT is not a *context* and ignored it. To reduce such incidences it is important to keep the number of background matches low which is achieved by using patterns with 18 match positions. However, as mentioned in the word count section, this also reduces the signal which makes their program better for closely related genomes.
The implementation of *CO-phylog* is as simple as standard word count methods. Similar to other word count methods, spaced-word matches have to be identified first and then it is straight forward to determine the fraction of shared or different *objects*. Therefore,

CO-phylog has the same time and space complexity as other word count methods. Finally, it is notable that *CO-phylog* can work with unassembled reads.

In the following, I point out the differences between standard word count methods and *CO-phylog*. The major difference is that *CO-phylog* does not define distances based on word frequency vectors as standard word count methods but estimates distances directly from spaced-word matches. That means they consider spaced-word matches explicitly as short local gap-free alignments. If spaced-word matches are interpreted as alignments which represent homologies then is obvious that the mismatch rate can be directly estimated based on the aligned nucleotides at the *don't care* positions. However, one might wonder why methods are called alignment-free if they defines distances based on micro-alignments. The reason is that these methods do not align sequences over their entire length and also do not include gaps. Therefore, they are as fast and versatile as other alignment-free methods.

The second approach, *andi* [38] which stands for *ANchor DIstance*, identifies micro-alignments based on pairs of exact maximal unique matches, called *anchors*. The mismatch rate is calculated based on the aligned nucleotides between those anchors. An example is shown below

S_1	A	T	A	**C**	**A**	**T**	**G**	**T**	**A**	**G**	**G**	**C**	G	G
S_2		G	C	**C**	**A**	**T**	**C**	**T**	**C**	**G**	**G**	**C**	T	A	A

In this example, two anchors are found: CAT and GGC, colored in blue. Both are unique and maximal, i.e. they occur only once in each sequence and they can not be extended to the left or right without mismatches. The segment between both anchors, colored in red, has the same length in both sequences, i.e. they are equidistant, and the mismatch rate can be directly calculated based on the aligned nucleotides of this segment. However, the segments can have different lengths as shown in the example below:

S_1	A	T	A	**C**	**A**	**T**	**G**	**T**	**A**	**A**	**G**	**G**	**C**	G	G
S_2		G	C	**C**	**A**	**T**	**C**	**T**	**C**	**G**	**G**	**C**	T	A	A

There are two reasons why this can happen: either the segment between the anchors contain indels or at least one anchor occurred by chance and not due to homology. In both cases

no meaningful distance can be calculated and the authors ignore such segments. However, the chance of random anchors increases with increasing sequence divergence and also two non homologous anchors can be equidistant to each other. In this case, the mismatch rate between unrelated parts of the sequences are calculated which distorts the estimated distance. To reduce the chance of random anchors, the authors defined a minimum anchor length. Instead of a fixed minimal anchor length, they derived the probability density distribution of anchor lengths for unrelated sequences and defined the minimum length to be the 97.5% quantile of this distribution. The implementation of *andi* is based on *enhanced suffix arrays* and *range minimum queries* [27]. Compared to other alignment-free programs, the run time and the memory requirement of *andi* is very low. In fact, *andi* is one of the fastest alignment-free method available.

Andi and *CO-phylog* share the same underlying idea to estimate distances based on short local gap-free alignments. The difference between both methods is how the segments, from which the distances are estimated, are found. *CO-phylog* searches for spaced-word matches which have a predefined fixed length and one single wildcard character in the middle of the match. *Andi*, on the other hand, aligns segments of varying lengths and can be seen as a generalization of *CO-phylog*. Both methods determine the fraction of the mismatches of the aligned segments to estimate the number of mismatches. *Andi* turns this mismatch rate into an evolutionary distances by using the *Jukes&Cantor* model. The distances are estimated very accurately up to 0.5 substitutions per site. For higher substitution rates no anchors are found anymore that fulfill the minimal length or equidistant criteria. *CO-phylog*, on the other hand, does not correct for back substitutions, i.e. no substitution model is used to turn the mismatch rate into an evolutionary distance. Therefore, these distances rather reflect the p-distance than the number of substitutions per site.

Methods which calculate distances based on micro-alignments have some favorable properties over word count and match lengths methods. They estimate the number of substitutions between sequences without a simplified model of evolution which rely on unrealistic a priori assumptions such as that the sequences under study are related over their entire length. For example, the estimator based on Equation 1 assumes that for each position in the first sequence, there is exactly one match in the other sequence due to homology. However, if the sequences are only partially related, i.e. they only share local homologies, fewer word matches are found which leads to an overestimation of the substitution rate.

CO-phylog, on the other hand, is not dependent on the *number* of word matches because the distances are estimated from the aligned nucleotides at the *don't care* position. If one assumes that no or only few word matches are found in unrelated parts of the sequence, then the substitution rate is well approximated. But here is the problem: how to ensure that most spaced-word matches reflect homologies and not background matches? Both, *andi* and *CO-phylog* solve this issue by using long exact word matches to reduce background hits. However, as pointed out before, this also reduce the signal from which the distances are estimated. Consequently, both programs work well for smaller substitution rates but reach their limits if more divergent sequences are to be analysed.

1.2 Objectives and overview

The objective of this work is the development of a new alignment-free method for whole genome phylogeny reconstruction. The new approach should estimate the number of substitutions per position without sequence alignment. Current alignment-free estimators are either based on unrealistic a priori assumptions or are limited to closely related sequences. The new approach must overcome these limitations and an efficient implementation is necessary to keep the run time low.
In the first publication (see Chapter 2) a new approach to alignment-free whole genome phylogeny reconstruction, called *filtered spaced-word matches (FSWM)* is introduced. This approach is based on micro-alignments and uses a filtering procedure to remove spaced-word matches that are assumed not to be homologous.
The second publication (see Chapter 3) describes how the filtered spaced-word matches can be used as anchor points in genome alignments. To this end, a modified version of the *FSWM* approach was integrated into the multiple-sequence-alignment pipeline mugsy [6].
In Chapter 4, the *Prot-SpaM* approach is introduced. *Prot-SpaM* is a modification of *FSWM* and designed to calculate pairwise distances between whole proteoms. The calculated distances approximate the *PAM* distance [19] and is the first alignment-free program that estimates evolutionary distances between protein sequences without sequence alignments.
The overall results of the publications are discussed in Chapter 5 and an outlook is given in Chapter 6 which describes how the *FSWM* approach could be developed further.

2 Fast and accurate phylogeny reconstruction using filtered spaced-word matches

Reference

Chris-Andre Leimeister, Salma Sohrabi-Jahromi, and Burkhard Morgenstern. Fast and accurate phylogeny reconstruction using filtered spaced-word matches. Bioinformatics, 33:971-979, 2017. [61]

Original Contribution

CAL designed the study, developed and implemented *FSWM* and performed most of the program evaluations. SSJ contributed to the program evaluation. BM guided the work. CAL and BM wrote the Manuscript. All authors read and approved the final version of the manuscript.

Bioinformatics, 33(7), 2017, 971–979
doi: 10.1093/bioinformatics/btw776
Advance Access Publication Date: 4 January 2017
Original Paper

OXFORD

Phylogenetics

Fast and accurate phylogeny reconstruction using filtered spaced-word matches

Chris-André Leimeister[1,*], Salma Sohrabi-Jahromi[1] and Burkhard Morgenstern[1,2]

[1]Department of Bioinformatics, University of Göttingen, Institute of Microbiology and Genetics, Goldschmidtstr. 1, 37077 Göttingen, Germany and [2]University of Göttingen, Center for Computational Sciences, Goldschmidtstr. 1, 37077 Göttingen, Germany

*To whom correspondence should be addressed
Associate Editor: Alfonso Valencia

Received on August 2, 2016; revised on November 9, 2016; editorial decision on November 30, 2016; accepted on December 2, 2016

Abstract

Motivation: Word-based or 'alignment-free' algorithms are increasingly used for phylogeny reconstruction and genome comparison, since they are much faster than traditional approaches that are based on full sequence alignments. Existing alignment-free programs, however, are less accurate than alignment-based methods.

Results: We propose *Filtered Spaced Word Matches (FSWM)*, a fast alignment-free approach to estimate phylogenetic distances between large genomic sequences. For a pre-defined binary pattern of *match* and *don't-care* positions, *FSWM* rapidly identifies *spaced word-matches* between input sequences, i.e. gap-free local alignments with matching nucleotides at the *match* positions and with mismatches allowed at the *don't-care* positions. We then estimate the number of nucleotide substitutions per site by considering the nucleotides aligned at the *don't-care* positions of the identified spaced-word matches. To reduce the noise from spurious random matches, we use a filtering procedure where we discard all spaced-word matches for which the overall similarity between the aligned segments is below a threshold. We show that our approach can accurately estimate substitution frequencies even for distantly related sequences that cannot be analyzed with existing alignment-free methods; phylogenetic trees constructed with *FSWM* distances are of high quality. A program run on a pair of eukaryotic genomes of a few hundred Mb each takes a few minutes.

Availability and Implementation: The program source code for FSWM including a documentation, as well as the software that we used to generate artificial genome sequences are freely available at http://fswm.gobics.de/

Contact: chris.leimeister@stud.uni-goettingen.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Phylogeny reconstruction is one of the most fundamental tasks in computational biology. Traditionally, phylogenetic trees are inferred from multiple sequence alignments, by considering substitutions that may have occurred since the aligned sequences have evolved from a hypothetical common ancestor. While this procedure is still standard in phylogeny analysis, approaches based on *word statistics* have become popular in recent years, since they circumvent various difficulties involved in multiple alignment (Bernard *et al.*, 2016; Reinert *et al.*, 2009; Song *et al.*, 2014; Vinga, 2014; Wan *et al.*, 2010). The main advantage of these methods is that they are much faster than alignment-based approaches. Under most scoring schemes, calculating an optimal alignment of two sequences takes time proportional to the product of their lengths and is therefore limited to rather short sequences. By contrast, the word composition of sequences can be calculated in linear time. Another difficulty with

traditional phylogeny approaches is that sets of orthologous genes must be identified first, before multiple alignments can be calculated. Word-based methods, on the other hand, can be directly applied to genomic sequences, and even to unassembled reads. Since these approaches do not require global alignments of the sequences under study, they are often called *alignment free*. Strictly spoken, this is not quite correct, as most word methods compare—i.e. align—subwords of the sequences to each other. We use the term *alignment-free* anyway, since it is now commonly used for word-based approaches to sequence comparison.

Alignment-free methods are not only used in phylogenetic studies (Bromberg *et al.*, 2016; Didier *et al.*, 2007; Hatje and Kollmar, 2012), but also for *protein classification* (Comin and Verzotto, 2011; Leslie *et al.*, 2002; Lingner and Meinicke, 2006, 2008), *read alignment* (Ahmadi *et al.*, 2011; Langmead *et al.*, 2009; Li *et al.*, 2008), *isoform quantification from RNAseq reads* (Patro *et al.*, 2014), *sequence assembly* (Zerbino and Birney, 2008), *metagenomics* (Chatterji *et al.*, 2008; Leung *et al.*, 2011; Meinicke, 2015; Tanaseichuk *et al.*, 2012; Teeling *et al.*, 2004; Wang *et al.*, 2012; Wu and Ye, 2011), *analysis of regulatory elements* (Federico *et al.*, 2012; Kantorovitz *et al.*, 2007; Leung and Eisen, 2009; Wang *et al.*, 2012) and to identify *biomarkers in diagnostic tests* (Drouin *et al.*, 2016). Most authors divide alignment-free approaches into two classes: methods based on *word count* and methods based on *match lengths* (Haubold, 2014). For a fixed word length, word-count methods transform the input sequences into word-frequency vectors; the distance between two sequences can then be defined as the distance between the corresponding word-frequency vectors, for example under the Euclidean norm (Chor *et al.*, 2009; Sims *et al.*, 2009; Vinga *et al.*, 2012; Zuo and Hao, 2015).

Match-length approaches, in contrast, estimate phylogenetic distances from the length of substring matches between two sequences (Comin and Verzotto, 2012; Haubold *et al.*, 2005; Thankachan *et al.*, 2016; Ulitsky *et al.*, 2006). Since the length of exact substring matches between two homologous sequence regions depends on the mismatch frequency, substitution rates can be estimated, in turn, from the average length of exact common substrings (Domazet-Loso and Haubold, 2009). The program K_r (Haubold *et al.*, 2009) is based on this idea; to our knowledge, this was the first alignment-free approach that estimates phylogenetic distances based on an explicit model of molecular evolution.

Recently, we proposed to use so-called *spaced* words, instead of contiguous subwords of the input sequences, to quantify the similarity or dissimilarity between two sequences (Leimeister *et al.*, 2014). *Spaced* words are words containing wildcard characters at positions specified by a predefined binary pattern of *match* and *don't-care* positions. The main advantage of spaced words, compared to contiguous words, is that occurrences of neighbouring spaced words are statistically less dependent on each other; we have shown that better phylogenies can be obtained if *spaced*-word frequencies are used instead of the contiguous word frequencies used by traditional word-based methods (Horwege *et al.*, 2014; Leimeister *et al.*, 2014). As with most word-based methods, however, distances calculated from spaced-word-frequency vectors are not based on stochastic models of evolution; they do not try to estimate the 'true' distance between two sequences in a rigorous way, but provide only a rough measure of dissimilarity between the compared sequences.

Three other word-based methods have been proposed in recent years to estimate the mismatch frequency or number of substitutions per site between DNA sequences, namely *Co-phylog* (Yi and Jin, 2013), *andi* (Haubold *et al.*, 2015) and an estimator that is based on the *number* of *spaced* word matches between two sequences (Morgenstern *et al.*, 2015). *Co-phylog* uses so-called *micro alignments*, consisting of a single pair of aligned nucleotides, flanked on both sides by exact word matches of a fixed length ℓ. With our notation, a *micro alignment* can be seen as a match between two identical *spaced-words* of length $2\ell + 1$ with a single wildcard character at the middle position. To estimate the mismatch frequency between two sequences, *Co-phylog* calculates the fraction of *micro alignments* where the middle position is a mismatch.*andi* searches for pairs of maximal unique word matches within a certain distance to each other, and on the same diagonal in the comparison matrix of two sequences. The program then uses the implied gap-free alignments of the sequence segments between these word matches to estimate the number of substitutions per position. This can be seen as a generalization of *Co-phylog*, with more than one wildcard character in the middle, and with flanking word matches of varying length. Finally, we proposed in a previous paper to estimate evolutionary distances based on the number of (spaced) word matches between the sequences (Morgenstern *et al.*, 2015). This approach is more accurate than other alignment-free approaches. It is limited, however, to homologies extending over the full length of the input sequences, therefore this previous approach cannot be applied to compare distantly related genomes.

To accurately estimate the number of substitutions per position between two sequences, programs such as K_r, *andi* and *Co-phylog* have to consider (spaced) word matches between *homologous* segments of the input sequences. In order to exclude random background matches, they use *cut-off* values for the length of the matching word pairs—or, with our terminology, for the number of *match* positions in the matched *spaced* words. A difficulty with this approach is that a high cut-off is necessary if long sequences are compared, since the number of *background* matches increases quadratically with the sequence length, while the number of *homologous* matches increases only linearly. Thus, with cut-off that is sufficiently high to reduce the noise of random similarities, many *homologous* word matches will be discarded as well, which reduces the amount of information available for phylogeny inference.

In this paper, we propose *filtered spaced-word matches (FSWM)*, an alternative alignment-free approach to estimate phylogenetic distances between DNA sequences. *FSWM* first identifies all matching spaced words between two sequences, with respect to a fixed pattern of *match* and *don't-care* positions. Similar to *Co-phylog* and *andi*, we look at the aligned nucleotides at the *don't-care* positions of those spaced-word matches to estimate the average number of substitutions per sequence position. The fundamental difference between our method and these earlier methods is the way we filter out random background spaced-word matches. Instead of using a high number of *match positions* in the underlying pattern, we define a similarity score for spaced-word matches, considering the similarity between aligned nucleotides at the *don't-care* positions, and we discard all spaced-word matches with a score below a certain threshold. The fraction of mismatches at the *don't care* positions of the remaining spaced-word matches is then used to estimate the number of substitutions per position since two sequences diverged from a common ancestral sequence.

Using simulated and real genomic sequences, we show that *FSWM* can accurately estimate phylogenetic distances between genomic sequences. If distance matrices produced by *FSWM* are used as input for *Neighbor-Joining*, accurate phylogenetic trees can be obtained, even for large, distantly-related sequences. Calculating the evolutionary distance between two bacterial genomes of 3.3 Mb each takes around 0.2 s with our approach; for a pair of eukaryotic genomes of 340 Mb each, the runtime is around 320 s.

2 Algorithm

2.1 Spaced words and spaced-word matches

To describe our algorithm, we are using the terminology from our previous papers (Leimeister and Morgenstern, 2014; Morgenstern *et al.*, 2015). For an alphabet Σ, a sequence S of length L and $0 < i \leq L$, $S[i]$ denotes the ith symbol of S. A (binary) *pattern* is a word over {0, 1}; a position k in a pattern P is called a *match position* if $P[k] = 1$, it is called a *don't-care position* if $P[k] = 0$. The number of match positions in a pattern is called its *weight*. If '*' is a 'wildcard' character, $* \notin \Sigma$, a *spaced word* with respect to a pattern P is a word s over $\Sigma \cup \{*\}$ of the same length as P, with $s[i] \in \Sigma$ if i is a match position of P and $S[i] = *$ if i is a don't-care position of P. We say that a spaced word s with respect to some pattern P occurs in a sequence S at position i if $s[k] = S[i + k - 1]$ for all match positions k of P.

For sequences S_1 and S_2 over Σ, with lengths L_1 and L_2, respectively, a pattern P of length ℓ, and positions i, j, $1 \leq i \leq L_1 - \ell + 1, 1 \leq j \leq L_2 - \ell + 1$, we say that there is *spaced-word match* between S_1 and S_2 at (i, j) with respect to P if the same spaced word s occurs at position i in S_1 and at position j in S_2. In other words, the requirement is that for all match positions k in P, one has $S_1[i + k - 1] = S_2[j + k - 1]$. Below is a spaced-word match between two *DNA* sequences S_1 and S_2 at (5, 2) with respect to the pattern $P = 1100101$:

S_1:	G	C	T	G	T	A	T	A	C	G	T	C
S_2:				G	T	A	C	A	C	T	T	A	T
P:				1	1	0	0	1	0	1		

By definition, nucleotides in S_1 and S_2 corresponding to a *match position* of P are identical, while at the *don't-care* positions mismatches are possible. Throughout this paper, we use a *single* pattern P if two sequences are compared, as opposed to the *multiple-pattern* approach that we previously used (Leimeister *et al.*, 2014).

If one wants to estimate phylogenetic distances between genomic sequences based on spaced-word matches between them, one needs to distinguish between matches representing *true homologies* and random *background* matches (Devillers and Schbath, 2012). One possible way of reducing the number of background spaced-word matches would be to use a sufficiently high weight w, i.e. number of match positions, for the underlying pattern. Such an approach has been taken, for example, by *andi* and *Co-phylog*. For long, divergent input sequences, however, this approach is problematic. To see this, consider two sequences of length L under a model of evolution without insertions and deletions (indels), with a match probability p for pairs of homologous nucleotides and a background match probability q. With a pattern of length ℓ and weight w, the expected number of *homologous* spaced-word matches would be $(L - \ell + 1) \cdot p^w$, while the expected number of *background* matches would be $(L - \ell) \cdot (L - \ell + 1) \cdot q^w$. That means that, in order to obtain N times as many homologous spaced-word matches than background matches, one would have to use a weight w satisfying

$$w \geq \log_{\frac{p}{q}} [N \cdot (L - \ell)]$$

For two sequences of length 5 Mb, for example, with $p = 0.8$ and $q = 0.25$, a weight of $w = 16$ would be necessary to keep the fraction of background spaced-word matches below 10 % ($N = 9$); in this case, one would obtain around 140 000 homologous spaced-word matches and around 5800 background matches. By contrast, with the same L and q, but with $p = 0.6$, a weight of $w = 21$ would be necessary to have < 10% background matches. With these parameter values, as few as 114 expected spaced-word matches would be left as a basis for phylogeny reconstruction, 109 homologous and 5 background matches. With $p = 0.5$, it would be unlikely to find even a single spaced-word match with this approach.

2.2 Filtered spaced-word matches

Herein, we propose an alternative solution to distinguish between homologous and background spaced-word matches as a basis of phylogeny reconstruction. To identify all spaced-word matches between two sequences with respect to a pattern P, we sort the spaced words in the sequences lexicographically, such that matching spaced-words appear next to each other. A score is calculated for each spaced-word match using the following substitution matrix (Chiaromonte *et al.*, 2002)

	A	C	G	T
A	91	−114	−31	−123
C		100	−125	−31
G			100	−114
T				91

Here, we define the score of a spaced-word match as the sum of the substitution scores of the nucleotide pairs aligned at the *don't-care* positions. The spaced-word match shown in the previous section, for example, has three *don't-care positions* where the nucleotide pairs $(T, C), (A, A)$ and (G, T) are aligned; the score of this spaced-word match would thus be $-31 + 91 - 114 = -54$. Our algorithm discards all spaced-words matches with scores below a certain cut-off. Experimental results show that a cut-off value of zero is adequate to filter out most background similarities, see Table 1 and Figure 1, so our software uses this value by default.

For a sequence pair and a pattern P, one can plot the number of spaced-word matches against the similarity scores, i.e. for each possible score value, one plots the number of spaced-word matches with this score. We call such a plot a *spaced-words histogram*, examples are given in Figure 1. Under an *i.i.d.* model of molecular evolution, the scores of both *homologous* and *background* spaced-word matches are approximately normally distributed, with mean values $(\ell - w) \cdot s_h$ and $(\ell - w) \cdot s_b$, respectively, where s_h and s_b are the expected substitution scores for homologous and background nucleotide pairs. If, in addition, we consider a model without insertions and deletions, a spaced-word match at (i, j) is 'homologous' if and only if $i = j$, and each spaced-word match is either completely homologous or completely background. In this case, a *spaced-words histogram* is approximately the sum of two normal distributions. Figure 1 shows that, for real-world sequences too, the background spaced-word matches are roughly normally distributed. The distribution of the homologous spaced-word matches is more complex,

Table 1. Proportion of 'homologous' spaced-word matches retained after our 'filtering procedure'—i.e. after discarding all spaced-word matches with scores smaller or equal than zero—for gap-free simulated sequence pairs of different length and with 0.2–1.0 substitutions per site

	0.2	0.4	0.6	0.8	1.0
5 mb	1.00000	0.99998	0.99986	0.99769	0.98723
50 mb	0.99994	0.99950	0.99599	0.97795	0.86791
100 mb	0.99989	0.99898	0.99258	0.95765	0.79022

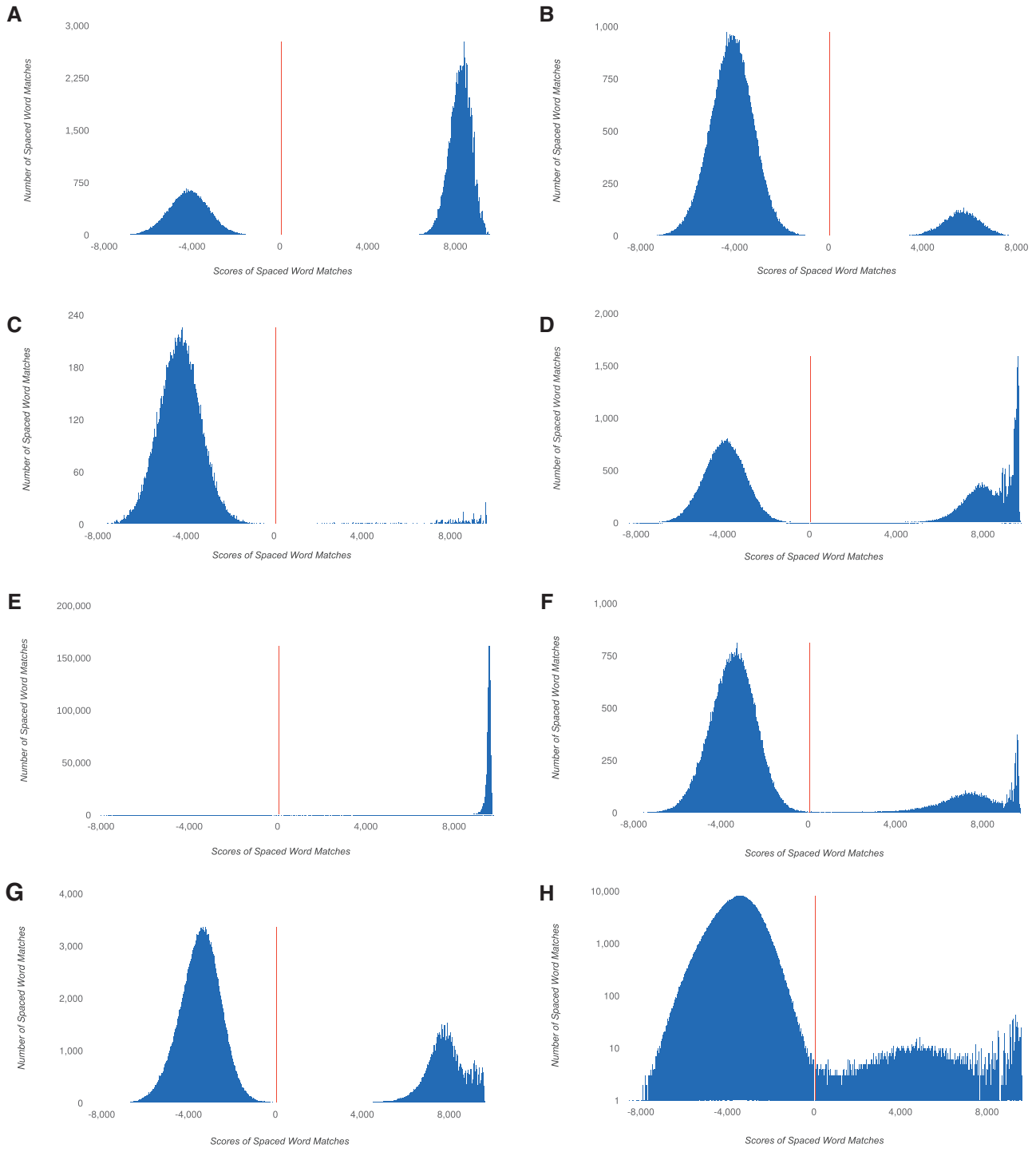

Fig. 1. *Spaced-word histograms* for simulated and real-world sequence pairs. The number of spaced-word matches is plotted against the spaced-word score as defined in the main text. The plots show the remaining spaced-word matches *after* the greedy one-to-one mapping explained in the main text. Thus, a spaced word at a certain position can be involved in at most one spaced-word match. (**A**) simulated indel-free sequence pairs of length 5 mb under an *i.i.d.* substitution model with a transition/transversion ratio of 2:1 and 0.1 substitutions per sequence position on average; (**B**) same model with 0.3 substitutions per position; (**C**) *Sagittula stellata* E37 vs *Rhodobacterales bacterium* HTCC2255; (**D**) *Octadecabacter arcticus* 238 vs *Octadecabacter antarticus* 307; (**E**) *Escherichia coli* strain S88 vs *Escherichia coli* strain 536; (**F**) *Phaeobacter gallaeciensis* 2.10 vs *Rhodobacterales bacterium* Y4I, (**G**) *Saccharomyces mikatae vs Saccharomyces cerevisiae*, (**H**) *Spizellomyces punctatus vs Batrachochytrium dendrobatidis*. For all sequence pairs, the scores of the background spaced-word matches are approximately normally distributed. For the real-world sequences, the peaks of the homologous matches are more complex, due to varying degrees of sequence conservation within the genomes. In E, the background peak is not visible since the two *E.coli* genomes are so closely related that there are much more homologous than background spaced-word matches. In **H**, we used a logarithmic scale because, for these two sequences, there are many more background than homologous spaced-word matches and the homologous peak would not be visible with a linear scale. For all sequence pairs, we used a pattern P with the default weight of $w = 12$ and 100 *don't-care* positions, so the pattern length was 112

however, reflecting different degrees of sequence similarity in different parts of the sequences.

A well-known problem in phylogenomics are duplications in genomes, since only *orthologous* sequences can be used for phylogeny reconstruction (Huerta-Cepas *et al.*, 2016; Schreiber *et al.*, 2009; Waterhouse *et al.*, 2013). We address this issue by selecting a one-to-one matching of spaced words when comparing two sequences. If a spaced word s occurs at m positions in the first sequence and at n positions in the second sequence, there are $m \times n$ spaced-word matches involving s. To find a one-to-one mapping between the occurrences of s, we use a greedy approach: after our filtering procedure, we sort the remaining spaced-word matches according to their similarity scores, we pick the one with the highest score and remove the corresponding two occurrences of s from our list. Next, we select the highest scoring one among the remaining spaced-word matches etc. By picking high-scoring spaced-word matches first, we increase the probability of matching orthologous segments of the compared genomes.

As an example, consider the two sequences below and the pattern $P = 10\,011$.

S_1: G G A T A G G G T A T A T T A

S_2: A G G G T A A C G G A T A T

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Here, the spaced word $s = G{*}{*}TA$ occurs three times in S_1, at positions 1, 6 and 8, and twice in S_2, at positions 2 and 9, so we obtain 6 spaced-word matches involving s, namely at (1,2), (1,9), (6,2), (6,9), (8,2) and (8,9). The one at (8,2) has a negative score, as it aligns nucleotide pairs (T, G) and (A, G) at the *don't-care* positions, the same is true for the one at (8,9) that aligns (T, G) and (A, A). Thus, with our default cut-off value of zero, these two spaced-word matches will be discarded in our initial filtering procedure. To obtain a one-to-one mapping, we sort the remaining four spaced-word matches involving s according to their scores. Here, the one at (6,2) would be selected first as it aligns two nucleotide pairs (G, G) at the *don't-care* positions, so it would have a score of $100+100=200$. Next, the one at (1,9) would be selected that aligns (G, G) and (G, A), with a score of $100+91=191$. The third and fourth spaced-word match would be at (1,2) and (6,9), respectively, both aligning (G, G) and (A, G), so each one would have score of $100 - 31 = 69$. We do not accept them in our one-to-one matching, however, since these occurrences of the spaced word s have already been used in the previously accepted, higher-scoring spaced-word matches.

After a set of spaced-word matches has been selected for a pair of genomic sequences as described, we estimate the evolutionary distance between the sequences by considering all *don't-care* positions of these spaced-word matches. From the aligned nucleotides at these positions, we estimate the match probability p and apply the usual *Jukes-Cantor* correction (Jukes and Cantor, 1969) to estimate the average number of substitutions per sequence position. Note that the spaced words that are finally selected can overlap so, in theory, a position in one sequence can be assigned to up to $\ell - w$ positions in the second sequence when p is estimated. For the above shown sequence pair, for example, there would be an additional spaced-word match with a positive score, namely the one at (7, 10) involving the spaced word $G{*}{*}AT$. As a result, the G at position 7 in S_1 would be assigned by two different spaced-word matches—the ones at (6, 2) and (7, 10)—to two different positions in S_2, positions 3 and 11. In the interest of program runtime, we do not remove such double assignments.

The runtime of our program depends on the number of spaced-word matches between the input sequences with grows quadratically with the sequence length. Since matches involving the same spaced word s are sorted to obtain a one-to-one matching, the *worst-case* complexity of our algorithm is $O(L^2 \cdot \log L)$. For realistic data, however, this worst-case estimate is hardly relevant, since the *real* number of spaced-word matches is only a tiny fraction of the theoretical maximum. Moreover, in real-world sequences not too many spaced words appear more than once, and only small sets of spaced-word matches need to be sorted for the greedy one-to-one matching. To further decrease the runtime for very long genomes, the weight w of the underlying pattern can be increased, to decrease the number of spaced-word matches. The runtime of our program on real-world and simulated sequences is reported in the next section. In addition to the weight w, the user can adjust the threshold for the spaced-word matches in the filtering procedure which is, by default, set to zero. By contrast, the number of *don't-care* positions is hard-coded in the current implementation, we use patterns with 100 don't care positions. With our default value of $w = 12$, spaced words have therefore a length of $\ell = 112$.

3 Test results

To evaluate the accuracy of the evolutionary distances estimated with *FSWM*, we performed systematic test runs on simulated and on real-world genomes. In all these test runs, we used the default weight $w = 12$ and a threshold of zero for the spaced-word scores in the filtering procedure. With some sequences we did additional test runs with alternative values of w. Binary patterns were generated with our software tool *rasbhari* (Hahn *et al.*, 2016).

3.1 Simulated sequences

As a first set of test data, we generated semi-artificial sequence pairs. Here, we used the genome sequence of *E. coli, strain K12*, as ancestral sequence and evolved it into pairs of descendant synthetic genomes by randomly generating an average number of d substitutions per site; we varied d between 0 and 1 in steps of 0.05 and used a transition/transversion ratio of 2:1. For each value of d, we generated 500 pairs of simulated genomes, estimated their distances with the methods under study and computed the standard deviations of the estimated distances. For a first set of sequence pairs, we did not include insertions and deletions. To make the simulation more realistic we generated a second set of sequence pairs where insertions and deletions were included with a probability of 0.5% at every position. The length of indels was randomly chosen between 1 and 100 with uniform probability. In Figure 2, the distances estimated with *Co-phylog, andi* and *FSWM* for these simulated sequence pairs, with and without indels, are plotted against the corresponding 'real' distances, i.e. the average number d of substitutions per site used to generate them. As mentioned, we used the default weight of $w = 12$, but with other values for w, similar results were achieved.

As can be seen in In Figure 2, *FSWM* estimates phylogenetic distances accurately for distances up to around 0.85 substitutions per position; for larger substitution rates, distances are slightly underestimated. The distance estimates of the program are hardly affected by insertions and deletions in the sequences. *andi*, by comparison, returns accurate distances in the range up to around 0.6 substitutions per position for our indel-free sequence pairs; this confirms previous results published by the authors of the program who also used indel-free sequence pairs in their program evaluation (Haubold *et al.*, 2015). For sequences with insertions and deletions, however,

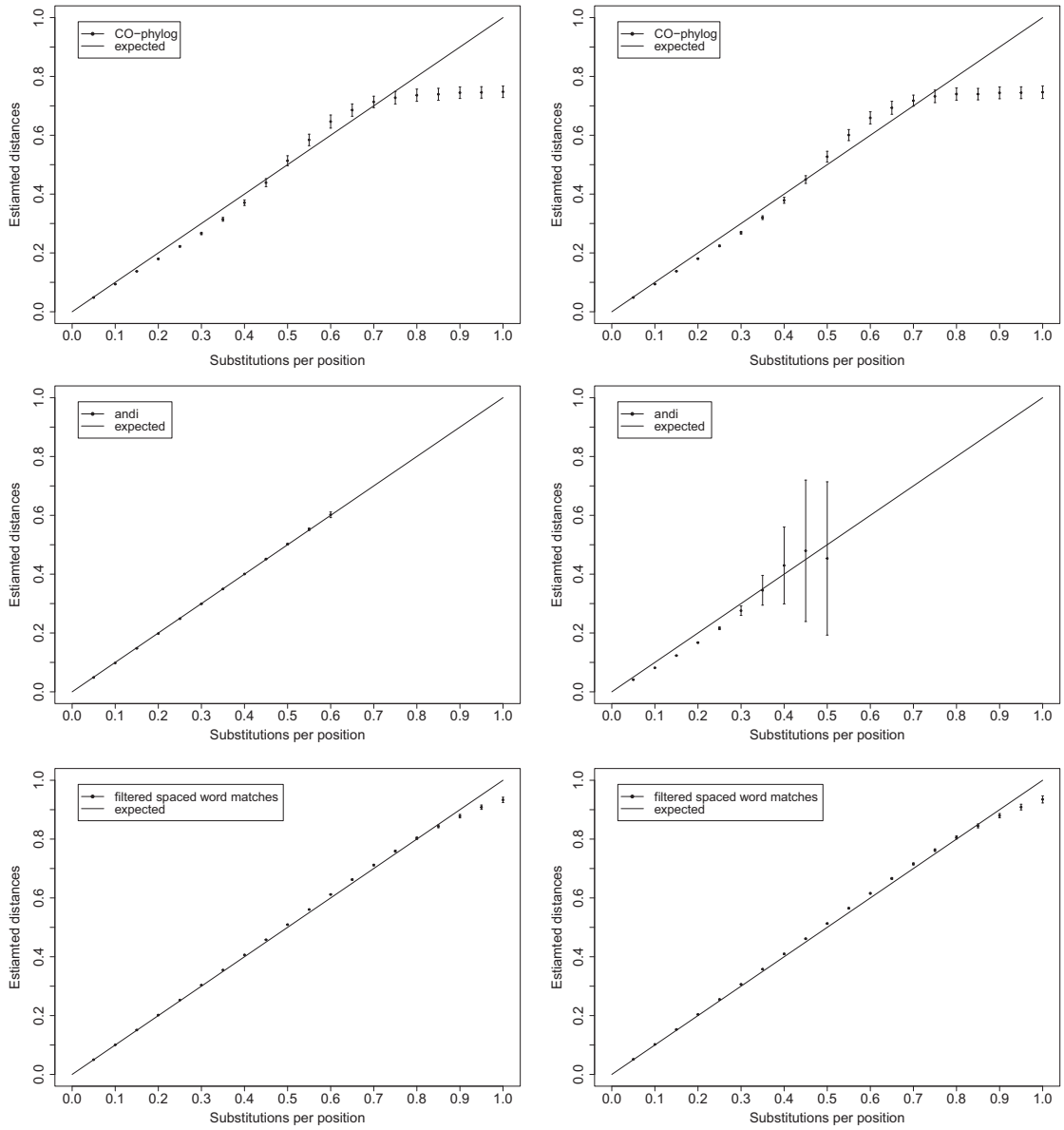

Fig. 2. Distances estimated with *Co-phylog* (top), *andi* (middle) and *FSWM* (bottom) for pairs of simulated DNA sequences, without indels (left-hand side) and with indels (right-hand side), plotted against the 'real' distances, measured in substitutions per site. Sequence pairs were generated as explained in the main text, Section 3.1, by inserting random mutations into the *E. coli K12* genome sequence. Error bars represent standard deviations

the results of the program are reliable only for distances up to around 0.35 substitutions per positions. *Co-phylog*, finally, produces reasonably good distance estimates in the range between 0 and around 0.75 substitutions per position, although this program is less accurate and produces statistically less stable results than *FSWM*. For larger substitution rates, the distances estimated by *Co-phylog* level out. As with *FSWM*, insertions and deletions hardly affect the performance of *Co-phylog* on these test data.

Next, we simulated sets of gene sequences with the *Artificial Life Framework (ALF)* developed by Dalquen *et al.* (2012). *ALF* evolves gene sequences based on a probabilistic model along a random tree, starting with a common ancestral sequence. During this process, evolutionary events are logged such that the 'true' phylogeny is known for each simulated sequence set and can be used as a reference in benchmark studies. We generated a series of 35 datasets containing 50 'species' each with a minimum gene length of 1000 and with default settings for all other parameters. Each dataset comprises 1500 simulated gene families with one gene for each of the 50 species, generated along the same tree. The total length of the sequences in one dataset is between 225 Mb and 463 Mb, the largest distance between two sequences in this dataset is around 0.4 substitutions per position.

For each dataset, we calculated distance matrices with *FSWM*, *Co-phylog* and *andi*. We then applied the *Neighbor Joining (NJ)* algorithm (Saitou and Nei, 1987) from the *PHYLIP* package (Felsenstein, 1993) to these distance matrices to calculate phylogenetic trees. Finally, we compared the obtained trees to the reference trees using the *Robinson-Folds (RF) metric* (Robinson and Foulds, 1981) to assess their quality. The smaller the *RF* distances are, the better are the reconstructed trees. The sum of the *RF* distances over all 35 datasets was 470 for the distances calculated by *andi*, 446 for the *Co-phylog* approach and 424 for our *FSWM* method.

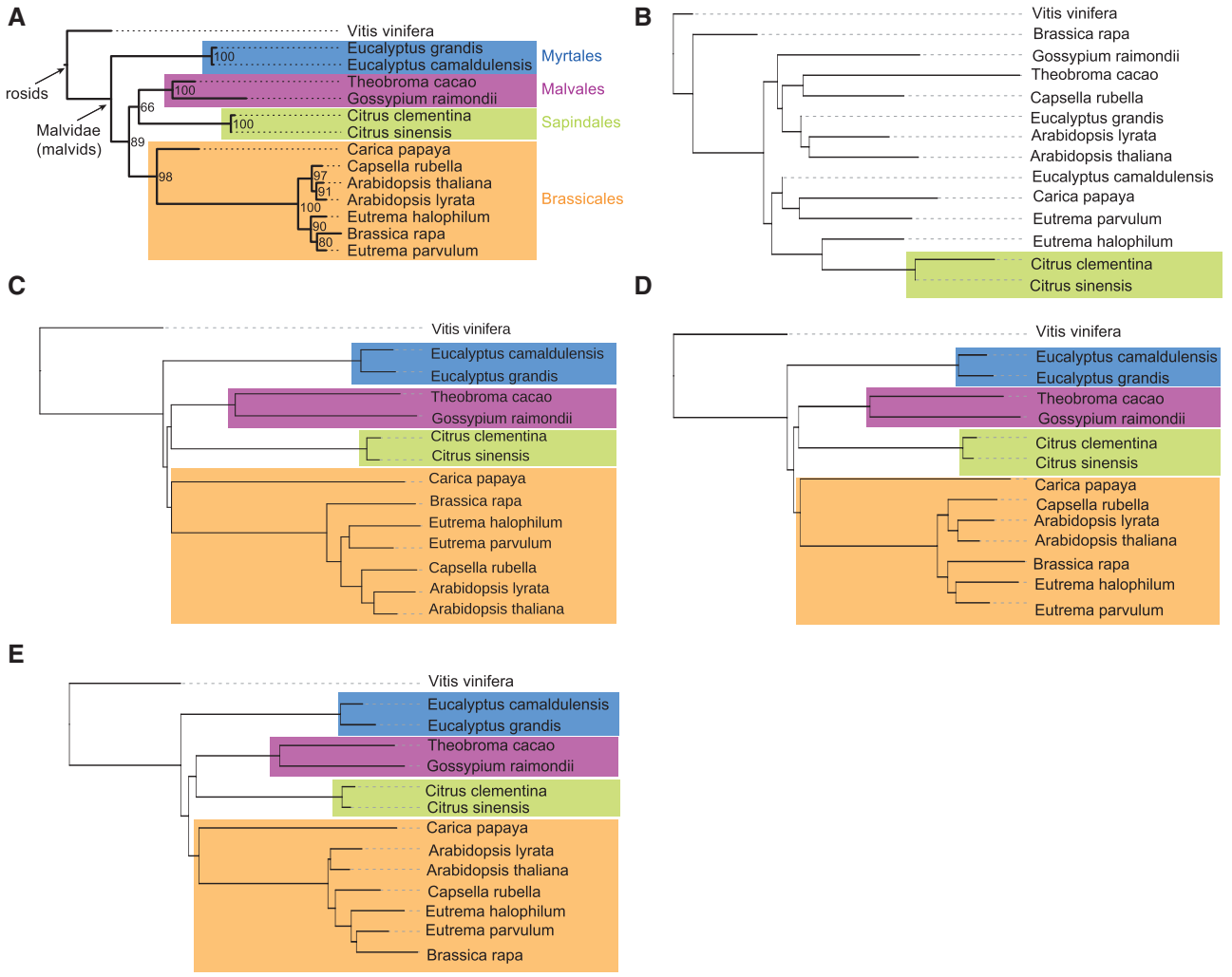

Fig. 3. Trees reconstructed from 14 plant genomes. (**A**) tree based on multiple protein alignments and *Maximum Likelihood* (Hatje and Kollmar, 2012); (**B**) tree calculated with distances from *andi*; (**C–E**) trees calculated with distances from *FSWM* with weights $w = 12$ (C), $w = 13$ (D) and $w = 14$ (E), respectively

3.2 Real genomes

To see if similar results can be achieved on real-world genomes, we first used a set of 13 bacterial genomes from the *Brucella* genus. As a reference, we used a tree that has been previously published by Foster *et al.* (2009) which is based on orthologous SNPs, discovered by the alignment program *MUMmer* (Kurtz *et al.*, 2004). The total size of this dataset is about 43.5 Mb; the 13 genomes are closely related, the largest distance between two genomes in this set is around 0.002 substitutions per position. All three programs, *Co-phylog*, *andi* and *FSWM*, precisely produced the topology of the reference tree, i.e. the *RF* distances between the reconstructed trees and the reference tree are all zero. For the pattern weight in *FSWM*, we used not only the default value of $w = 12$, but also $w = 10,11,13,14$. With all these values for w, we obtained exactly the same correct tree topology; these trees are shown in the *supplementary material*.

As a third benchmark set for phylogeny reconstruction, we used a set of 14 plant genomes with a total size of about 4.8 *Gb* which is frequently used as a test case in alignment-free studies (Hatje and Kollmar, 2012; Leimeister and Morgenstern, 2014). These sequences are rather distantly related, the maximum distance between two genomes in this set is 0.633 substitutions per position. Figure 3 shows a previously published tree that has been calculated using *Maximum Likelihood* based on manually improved multiple sequence alignments of *CAP* and *Arp2/3* protein sequences (Hatje and Kollmar, 2012), a tree obtained with *andi* and three trees obtained with *FSWM* using parameter values $w = 12$, $w = 13$ and $w = 14$.

The trees obtained with our approach are similar to the tree published by Hatje and Kollmar (2012), with only minor differences in the *Brassicales* clade: with $w = 12$ and $w = 13$, *Brassica rapa* has a slightly different position in the *FSWM* tree, compared to the tree based on protein alignments, while with $w = 14$, *Capsella rubella* is placed at a different position. *andi* did not produce a reasonable phylogeny for these genomes, since this program works best on sequences with lower substitution rates. We also tried to run *Co-phylog* on this dataset, but the program did not terminate, so we were unable to include its results in our evaluation. As reported in the literature, other alignment-free methods were also unable to calculate meaningful phylogenies for this dataset (Hatje and Kollmar, 2012; Leimeister and Morgenstern, 2014).

3.3 Runtime

We ran all programs on 10 x *Intel(R) Xeon(R) CPU E7-4850* with 2.00 GHz with 4 cores each summing up to 40 cores (80 threads)

and 1000 GB RAM. *Co-phylog* took around 1200 s for one of the simulated *ALF* datasets, *andi* 22 s and *FSWM* 180 s. For the *Brucella* genomes the runtime was 3 s for *andi*, 59 s for *Co-phylog* and 15 s for *FSWM*. For the plant genomes, the runtime was 1740 s for *andi*, for *FSWM* the runtime was 129 540 s with $w = 12$, compared to 28 980 s with $w = 13$ and 10 260 s with $w = 14$.

4 Discussion

In this paper, we proposed *Filtered Spaced-Word Matches (FSWM)*, a new alignment-free approach to estimate phylogenetic distances between genomic sequences. Similar to the recently published methods *andi* and *Co-phylog*, *FSWM* rapidly identifies pairwise local gap-free alignments where pairs of identical nucleotides are aligned to each other at certain, pre-defined positions, while mismatches are possible elsewhere. Phylogenetic distances between genomes can then be estimated by considering those positions of the identified local alignments where mismatches are allowed. While *andi* and *Co-phylog* use local alignments bounded by matching word pairs of a certain length, our approach uses *spaced-word* matches with respect to an arbitrary binary pattern of *match* and *don't-care* positions.

The main difference between *FSWM* and these previous methods is in how we distinguish between local homologies and spurious random similarities. *andi* and *Co-phylog* use exact word matches of a certain length to reduce the background noise. A disadvantage of this approach is that, this way, many true homologies are discarded as well. By contrast, we use patterns with a rather low number w of *match positions*, the default value in *FSWM* is $w = 12$. This allows us to identify sufficiently many local homologies, even for remotely related sequences. To filter out random similarities, we then look at the nucleotides aligned at the *don-t care* positions, and we discard all spaced-word matches for which the overall similarity is below a certain threshold. We use patterns with 100 *don't care* positions, so the default length of our spaced-word matches is 112 *nt*. To deal with duplications, we select a one-to-one mapping of spaced words from the compared sequences.

Our approach is able to rapidly detect homologies among genomic sequences, as a basis for phylogeny reconstruction. At the same time, our filtering procedure allows us to distinguish between true homologies and spurious random similarities. This way, *FSWM* can accurately estimate substitution frequencies, even for long, distantly related sequences where established alignment-free methods fail to produce reasonable results.

For closely related sequences, our filtering approach can separate homologous spaced-word matches from background matches with almost 100% accuracy. For distantly related sequences, there is a certain twilight zone where the distributions of the homologue and background matches in the spaced-word histograms have some overlap, as can be seen in the comparison of *Spizellomyces punctatus* and *Batrachochytrium dendrobatidis* in Figure 1H. If longer patterns with more *don't-care* positions would be used, the split between homologous and background spaced-word matches would become clearer, but the pattern length cannot be too long because this would reduce the number of spaced-word matches in homologous regions too much.

In the current version of *FSWM*, the main parameter that is to be adjusted by the user is the *weight w* of the underlying binary pattern. By default, we are using a low weight to obtain sufficiently many 'candidate' spaced-word matches that are then filtered based on the similarity between the aligned segments. If large genomes are compared, it is advisable to increase w to reduce the number of 'candidate' spaced-word matches, since this decreases the program runtime. Note that the value of w has no systematic influence on the estimated distances; in our test runs we obtained similar distance values and phylogenetic trees with different values of w; see also the *supplementary material* to this paper.

To clearly separate homologous from background spaced-word matches in our filtering procedure, we are using a relatively high number of *don't care* positions; in our implementation, the number of 100 *don't care* positions is hard-coded. A certain disadvantage of this approach is that we miss homologies containing insertions or deletions since, by definition, spaced-word matches are gap-free local alignments. Therefore, input sequences for *FSWM* must be long enough to ensure that sufficiently many homologous spaced-word matches are found, even for remotely related input sequences with frequent indels.

To separate homologous from background spaced-word matches, a suitable threshold needs to be defined for the similarity between matching spaced words. If the chosen threshold is too low, too many random similarities are accepted, and our approach *overestimates* distances between compared sequences. If the threshold is too high, the noise is reduced, but this way, low-scoring homologous spaced-word matches are also discarded and distances are *underestimated*. In *FSWM*, we use a nucleotide substitution matrix and, by default, we discard all spaced-word matches for which the total score over all *don-t-care* positions is negative. With this cut-off criterion, our method is able to accurately estimate substitution frequencies even for highly divergent genomic sequences. For very large substitution rates, however, our method slightly underestimates phylogenetic distances, so it is possible that *FSWM* discards too many low-scoring homologies. More sophisticated statistical methods may be applied to better distinguish between true homologies and random similarities in our approach to further improve its accuracy.

Funding

S.S.-J. was supported by a grant from *International Max Planck Research School Molecular Biology*, Göttingen.

Conflict of Interest: none declared.

References

Ahmadi,A. *et al.* (2011) Hobbes: optimized gram-based methods for efficient read alignment. *Nucleic Acids Res.*, **40**, e41.

Bernard,G. *et al.* (2016) Alignment-free microbial phylogenomics under scenarios of sequence divergence, genome rearrangement and lateral genetic transfer. *Scientific Reports*, **6**, 28970.

Bromberg,R. *et al.* (2016) Phylogeny reconstruction with alignment-free method that corrects for horizontal gene transfer. *PLOS Comput. Biol.*, **12**, e1004985.

Chatterji,S. *et al.* (2008) *Research in Computational Molecular Biology: 12th Annual International Conference, RECOMB 2008, Singapore, March 30 – April 2, 2008. Proceedings*, pp. 17–28. Springer, Berlin. Heidelberg.

Chiaromonte,F. *et al.* (2002) Scoring pairwise genomic sequence alignments. In: Altman,R.B. *et al.* (eds) *Pacific Symposium on Biocomputing*. World Scientific, Singapore, pp. 115–126.

Chor,B. *et al.* (2009) Genomic dna k-mer spectra: models and modalities. *Genome Biol.*, **10**, R108.

Comin,M. and Verzotto,D. (2011) The irredundant class method for remote homology detection of protein sequences. *J. Comput. Biol.*, **18**, 1819–1829.

Comin,M. and Verzotto,D. (2012) Alignment-free phylogeny of whole genomes using underlying subwords. *Algorithms Mol. Biol.*, **7**, 34.

Dalquen,D.A. *et al.* (2012) Alf-a simulation framework for genome evolution. *Mol. Biol. Evol.*, **29**, 1115–1123.

Devillers,H. and Schbath,S. (2012) Separating significant matches from spurious matches in DNA sequences. *J. Comput. Biol.*, **19**, 1–12.

Didier,G. *et al.* (2007) Comparing sequences without using alignments: application to HIV/SIV subtyping. *BMC Bioinformatics*, **8**, 1.

Domazet-Loso,M. and Haubold,B. (2009) Efficient estimation of pairwise distances between genomes. *Bioinformatics*, **25**, 3221–3227.

Drouin,A. *et al.* (2016) Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons. *BMC Genomics*, **17**, 754.

Federico,M. *et al.* (2012) Direct vs 2-stage approaches to structured motif finding. *Algorithms Mol. Biol.*, **7**, 20.

Felsenstein,J. (1993) Phylip (phylogeny inference package), version 3.5 c.

Foster,J. *et al.* (2009) Whole-genome-based phylogeny and divergence of the genus brucella. *J. Bacteriol.*, **191**, 2864–2870.

Hahn,L. *et al.* (2016) *rasbhari*: optimizing spaced seeds for database searching, read mapping and alignment-free sequence comparison. *PLOS Comput. Biol.*, **12**, e1005107.

Hatje,K. and Kollmar,M. (2012) A phylogenetic analysis of the brassicales clade based on an alignment-free sequence comparison method. *Front. Plant Sci.*, **3**, 192.

Haubold,B. (2014) Alignment-free phylogenetics and population genetics. *Brief. Bioinf.*, **15**, 407–418.

Haubold,B. *et al.* (2015) andi: fast and accurate estimation of evolutionary distances between closely related genomes. *Bioinformatics*, **31**, 1169–1175.

Haubold,B. *et al.* (2009) Estimating mutation distances from unaligned genomes. *J. Comput. Biol.*, **16**, 1487–1500.

Haubold,B. *et al.* (2005) Genome comparison without alignment using shortest unique substrings. *BMC Bioinf.*, **6**, 123.

Horwege,S. *et al.* (2014) Spaced words and kmacs: fast alignment-free sequence comparison based on inexact word matches. *Nucleic Acids Res.*, **42**, W7–W11.

Huerta-Cepas,J. *et al.* (2016) eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic Acids Res.*, **44**, D286–D293.

Jukes,T.H. and Cantor,C.R. (1969) *Evolution of Protein Molecules*. Academy Press, New York.

Kantorovitz,M.R. *et al.* (2007) A statistical method for alignment-free comparison of regulatory sequences. *Bioinformatics*, **23**, i249–i255.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25–R10.

Leimeister,C.A. *et al.* (2014) Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics*, **30**, 1991–1999.

Leimeister,C.A. and Morgenstern,B. (2014) kmacs: the k-mismatch average common substring approach to alignment-free sequence comparison. *Bioinformatics*, **30**, 2000–2008.

Leslie,C. *et al.* (2002) The spectrum kernel: A string kernel for SVM protein classification. In: *Proceedings of the Pacific Symposium on Biocomputing*, vol. 7, pp. 566–575.

Leung,G. and Eisen,M.B. (2009) Identifying *cis*-regulatory sequences by word profile similarity. *PLOS One*, **4**, 1–11.

Leung,H.C.M. *et al.* (2011) A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. *Bioinformatics*, **27**, 1489–1495.

Li,R. *et al.* (2008) Soap: short oligonucleotide alignment program. *Bioinformatics*, **24**, 713–714.

Lingner,T. and Meinicke,P. (2006) Remote homology detection based on oligomer distances. *Bioinformatics*, **22**, 2224–2231.

Lingner,T. and Meinicke,P. (2008) Word correlation matrices for protein sequence analysis and remote homology detection. *BMC Bioinformatics*, **9**, 259.

Meinicke,P. (2015) UProC: tools for ultra-fast protein domain classification. *Bioinformatics*, **31**, 1382–1388.

Morgenstern,B. *et al.* (2015) Estimating evolutionary distances between genomic sequences from spaced-word matches. *Algorithms Mol. Biol.*, **10**, 5.

Patro,R. *et al.* (2014) Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.*, **32**, 462–464.

Reinert,G. *et al.* (2009) Alignment-free sequence comparison (I): statistics and power. *J. Comput. Biol.*, **16**, 1615–1634.

Robinson,D. and Foulds,L. (1981) Comparison of phylogenetic trees. *Math. Biosci.*, **53**, 131– 147.

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

Schreiber,F. *et al.* (2009) Orthoselect: a protocol for selecting orthologous groups in phylogenomics. *BMC Bioinf.*, **10**, 219.

Sims,G.E. *et al.* (2009) Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proc. Natl. Acad. Sci.*, **106**, 2677–2682.

Song,K. *et al.* (2014) New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Brief. Bioinf.*, **15**, 343–353.

Tanaseichuk,O. *et al.* (2012) Separating metagenomic short reads into genomes via clustering. *Algorithms Mol. Biol.*, **7**, 27.

Teeling,H. *et al.* (2004) TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. *BMC Bioinformatics*, **5**, 163.

Thankachan,S.V. *et al.* (2016) ALFRED: a practical method for alignment-free distance computation. *J. Comput. Biol.*, **23**, 452–460.

Ulitsky,I. *et al.* (2006) The average common substring approach to phylogenomic reconstruction. *J. Comput. Biol.*, **13**, 336–350.

Vinga,S. (2014) Editorial: alignment-free methods in computational biology. *Brief. Bioinf.*, **15**, 341–342.

Vinga,S. *et al.* (2012) Pattern matching through chaos game representation: bridging numerical and discrete data structures for biological sequence analysis. *Algorithms Mol. Biol.*, **7**, 10.

Wan,L. *et al.* (2010) Alignment-free sequence comparison (II): theoretical power of comparison statistics. *J. Comput. Biol.*, **17**, 1467–1490.

Wang,Y. *et al.* (2012) MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics*, **28**, i356–i362.

Waterhouse,R.M. *et al.* (2013) Orthodb: a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Res.*, **41**, D358–D365.

Wu,Y.W.W. and Ye,Y. (2011) A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *J. Comput. Biol.*, **18**, 523–534.

Yi,H. and Jin,L. (2013) Co-phylog: an assembly-free phylogenomic approach for closely related organisms. *Nucleic Acids Res.*, **41**, e75.

Zerbino,D.R. and Birney,E. (2008) Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.*, **18**, 821–829.

Zuo,G. and Hao,B. (2015) CVTree3 web server for whole-genome-based and alignment-free prokaryotic phylogeny and taxonomy. *Genomics Proteomics Bioinf.*, **13**, 321– 331.

3 Accurate multiple alignment of distantly related genome sequences using filtered spaced word matches as anchor points

Reference

Chris-Andre Leimeister, Thomas Dencker, and Burkhard Morgenstern. Accurate multiple alignment of distantly related genome sequences using filtered spaced word matches as anchor points. Bioinformatics, doi:10.1093/bioinformatics/bty592, in press. [62]

Original Contribution

CAL designed the study, integrated *FSWM* into the mugsy pipeline and performed all program evaluations. TD contributed to the implementation. BM guided the work. CAL and BM wrote the Manuscript. All authors read and approved the final version of the manuscript.

Bioinformatics, 2018, 1–8
doi: 10.1093/bioinformatics/bty592
Advance Access Publication Date: 10 July 2018
Original Paper

OXFORD

Genome analysis

Accurate multiple alignment of distantly related genome sequences using filtered spaced word matches as anchor points

Chris-André Leimeister[1,*], Thomas Dencker[1] and Burkhard Morgenstern[1,2,*]

[1]Department of Bioinformatics, Institute of Microbiology and Genetics and [2]Center for Computational Sciences, University of Goettingen, 37077 Goettingen, Germany

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on May 23, 2017; revised on October 3, 2017; editorial decision on June 29, 2018; accepted on July 9, 2018

Abstract

Motivation: Most methods for pairwise and multiple genome alignment use fast local homology search tools to identify *anchor points*, i.e. high-scoring local alignments of the input sequences. Sequence segments between those anchor points are then aligned with slower, more sensitive methods. Finding suitable anchor points is therefore crucial for genome sequence comparison; speed and sensitivity of genome alignment depend on the underlying anchoring methods.

Results: In this article, we use *filtered spaced word matches* to generate anchor points for genome alignment. For a given binary pattern representing *match* and *don't-care* positions, we first search for *spaced-word matches*, i.e. ungapped local pairwise alignments with matching nucleotides at the *match* positions of the pattern and possible mismatches at the *don't-care* positions. Those spaced-word matches that have similarity scores above some threshold value are then extended using a standard *X*-drop algorithm; the resulting local alignments are used as anchor points. To evaluate this approach, we used the popular multiple-genome-alignment pipeline *Mugsy* and replaced the exact word matches that *Mugsy* uses as anchor points with our spaced-word-based anchor points. For closely related genome sequences, the two anchoring procedures lead to multiple alignments of similar quality. For distantly related genomes, however, alignments calculated with our *filtered-spaced-word matches* are superior to alignments produced with the original *Mugsy* program where exact word matches are used to find anchor points.

Availability and implementation: http://spacedanchor.gobics.de

Contact: chris.leimeister@stud.uni-goettingen.de or bmorgen@gwdg.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The most fundamental task in biological sequence analysis is to *align* two or several nucleic-acid or protein sequences—either *globally*, over their entire length, or *locally*, by restricting the alignment to a single region of homology. Standard approaches to global alignment assume that the input sequences derived from a common ancestor, and that evolutionary events are limited to substitutions and small insertions and deletions. In this case, sequence homologies can be represented by *global sequence alignments*, that is, by inserting *gap characters* into the sequences such that evolutionarily related sequence positions are arranged on top of each other. Under most scoring schemes, calculating an *optimal* alignment of two sequences takes time proportional to the product of their lengths and is therefore limited to rather short sequences (Durbin *et al.*, 1998; Gotoh, 1982; Morgenstern, 2002; Needleman and Wunsch, 1970; Smith and Waterman, 1981).

With the rapidly increasing number of partially or fully sequenced genomes, alignment of *genomic* sequences has become an important field of research in bioinformatics, see Earl *et al.* (2014) for a recent review and evaluation of some of the most popular approaches. Here, the first challenge is the sheer size of the input sequences which makes it impossible to use traditional algorithms with quadratic run time. A second challenge is the fact that related genomes often share *multiple* local homologies, interrupted by non-conserved parts of the sequences where no significant similarities can be detected. This means that neither *global* alignment methods (Needleman and Wunsch, 1970) nor strictly *local* methods (Altschul *et al.*, 1990; Smith and Waterman, 1981) are appropriate to represent the homologies between entire genomes. Finally, homologies do not generally occur in the same relative order in different genomes, because of duplications and large-scale genome rearrangements. Since it is not possible, in general, to represent homologies among genomes in one single alignment, advanced genome aligners return alignments of so-called *Locally Collinear Blocks*, i.e. blocks of segments of the input sequences where orthologous genes appear in the same linear order.

Since the late 1990s, efforts have been made to a address the above issues, and many approaches to genome-sequence alignment have been published. One of the first multiple-alignment programs that could be applied to genomic sequences was *DIALIGN* (Morgenstern *et al.*, 1996, 2002). This program composes multiple alignments from chains of local pairwise alignments, and it does not penalize gaps; it is therefore able to align sequences where local homologies are separated by non-homologous regions. The program was initially not designed for large genomic sequences, though, and it is limited to sequences up to around $10\,kb$. Moreover, *DIALIGN* is not able to deal with duplications, rearrangements or homologies on different strands of the DNA double helix.

To align longer sequences, most programs for genomic alignment rely on some sort of *anchoring* (Huang *et al.*, 2006; Morgenstern *et al.*, 2006). In a first step, they use a fast local alignment method to identify high-scoring local homologies, so-called *anchor points*. Next, *chains* of such local alignments are calculated and, finally, sequence segments between the selected anchor points are aligned with a slower but more sensitive alignment method. For multiple sequence sets, either pairwise or multiple local alignments can be used as anchor points. A pioneering tool to find anchor points for genomic alignment is *MUMmer* (Delcher *et al.*, 1999); the current version of the program is considered the state-of-the-art in alignment anchoring (Kurtz *et al.*, 2004). MUMmer uses *maximal unique matches* as pairwise anchor points. The genome aligner *MGA*, by contrast, uses *maximal exact matches* involving *all* input sequences (Höhl *et al.*, 2002). Both *MUMmer* and *MGA* use *suffix trees* (Kurtz, 1999) and related data structures to rapidly identify the pairwise or multiple word matches. *MUMmer* and *MGA* can rapidly align entire bacterial genomes; *MUMmer* was also used in the *A. thaliana* genome project (The Arabidopsis Genome Initiative, 2000). However, since the number of exact word matches decreases with increasing evolutionary distances, these approaches are most useful if closely related genomes are to be compared, such as different strains of *E. coli*.

Other approaches to genome alignment are *OWEN* (Ogurtsov *et al.*, 2002), *AVID* (Bray *et al.*, 2003), *MAVID* (Bray and Pachter, 2003), *LAGAN and Multi-LAGAN* (Brudno *et al.*, 2003b), *CHAOS/DIALIGN* (Brudno *et al.*, 2003a), the *VISTA genome pipeline* (Dubchak *et al.*, 2009), *TBA* (Blanchette *et al.*, 2004) and *Mauve* (Darling *et al.*, 2004), see Dewey and Pachter (2006) and Batzoglou (2005) for review. All of these methods use anchor points, and most of them are able to deal with duplications and genome rearrangements. Some genome aligners use *statistical* properties of the sequences (Bradley *et al.*, 2009; Darling *et al.*, 2004); other methods are based on *graphs*, for example on *A-Bruijn graphs* (Raphael *et al.*, 2004) or on *cactus graphs* (Paten *et al.*, 2011). A further development of *Mauve*, called *progressiveMauve* (Darling *et al.*, 2010), uses palindromic *spaced seeds* (Darling *et al.*, 2006) instead of exact word matches as anchor points. Spaced seeds are used for sequence-analysis tasks such as database searching (Choi *et al.*, 2004; Ma *et al.*, 2002; Noé, 2017; Xu *et al.*, 2006), read mapping (Břinda *et al.*, 2015; David *et al.*, 2011; Langmead *et al.*, 2009; Noé *et al.*, 2010; Ounit and Lonardi, 2015), alignment-free sequence comparison (Leimeister *et al.*, 2014) or pathogen detection Deneke *et al.* (2017). Such pattern-based approaches are often superior to methods based on contiguous words or word matches, see for example Li *et al.* (2006). In *Mauve*, palindromic patterns are used to cover both DNA strands of the input sequences.

Mugsy (Angiuoli and Salzberg, 2011) is a popular software pipeline for multiple genome alignment. In a first step, this program uses *nucmer* (Kurtz *et al.*, 2004) to construct all pairwise alignments of the input sequences. *Nucmer*, in turn, uses *MUMmer* to find exact unique word matches which are used as alignment anchor points. An *alignment graph* is constructed from these pairwise alignments using the *SeqAn* software (Döring *et al.*, 2008), and *Locally Collinear Blocks* are constructed. Finally, a multiple alignment is calculated using *SeqAn:: TCoffee* (Rausch *et al.*, 2008). *Mugsy* has been designed to rapidly align closely related genomes, such as different strains of a bacterium. Here, it produces alignments of high quality. On more distantly related genomes, however, the program is often outperformed by other multiple aligners (Earl *et al.*, 2014).

Finding *anchor points* is the most important step in whole-genome sequence alignment. Here, a trade-off between *speed*, *sensitivity* and *precision* has to be made. A sufficient number of anchor points is necessary to reduce the run time of the subsequent, more sensitive alignment routine. Wrongly chosen anchor points, on the other hand, can substantially deteriorate the quality of the final output alignment. They may not only lead to misalignments of non-homologous parts of the sequences but may also prevent biologically relevant, true homologies from being aligned. Also, if the number of anchor points is too large, finding optimal chains of anchor points can become computationally expensive.

In this article, we apply the *filtered spaced word matches (FSWM)* approach (Leimeister *et al.*, 2017) to find pairwise anchor points for genomic alignment. We use a *hit-and-extend* approach where high-scoring spaced-word matches are used as *seeds*. More precisely, for a given binary pattern of length ℓ representing *match* and *don't care* positions, we identify *spaced-word matches*—i.e. pairs of length-ℓ segments from the input sequences with matching nucleotides at the *match* positions and possible mismatches at the *don't care* positions. For each such spaced-word match, we then calculate a similarity score, and we keep only those spaced-word matches that have a score above a certain threshold. These matches are then extended to gap-free alignments, similar as in *BLAST* (Altschul *et al.*, 1990). To evaluate the anchor points generated by our approach, we modified the *Mugsy* pipeline by using our anchoring procedure instead of the original anchor points in *Mugsy* that are based on exact word matches. For closely related input sequences, these two different anchoring procedures lead to alignments of similar quality. Our anchor points are clearly superior, however, if distal sequences are to be aligned, where most other alignment approaches either fail to produce meaningful alignments or require an unacceptable amount of time.

Through our website at http://spacedanchor.gobics.de, we provide the modified *Mugsy* pipeline with our anchoring approach, as a pipeline for genome-sequence alignment that can be readily installed. In addition, we provide a stand-alone version of our software, such that software developers can integrate our anchor points into their own sequence-analysis pipelines.

2 Results

2.1 Filtered spaced word matches

For a sequence S of length L over an alphabet Σ and $0 < i \leq L$, $S[i]$ denotes the ith symbol of S, and $|S|$ denotes the *length* of S. Throughout this article, a *pattern* is a word over $\{0, 1\}$. For a pattern P, a position i is called a *match position* if $P[i] = 1$ and a *don't-care positions* otherwise. The number of match positions in a pattern P is called the *weight* of P. For an alphabet Σ, a pattern P, and a wildcard character '*' not contained in Σ, a *spaced word* with respect to P is a word w over $\Sigma \cup \{*\}$, such that $w[k] = *$ if and only if k is a *don't-care position*, see also Leimeister *et al.* (2014) and Horwege *et al.* (2014). We say that a spaced word w with respect to a pattern P occurs in a sequence S at some position i, if $i \leq |S| - |P| + 1$, and if $S[i + k - 1] = w[k]$ for all match positions k of P.

For sequences S_1 and S_2, a pattern P, and positions i and j, we say that there is *spaced-word match* between S_1 and S_2 at (i, j) with respect to P if the same spaced word occurs at i in S_1 and at j in S_2—in other words, if for all match positions k in P, one has

$$S_1[i + k - 1] = S_2[j + k - 1].$$

For the two sequences S_1 and S_2 below, for example, there is a spaced-word match with respect to the pattern $P = 1100101$ at $(5, 2)$:

S_1 :	G	C	T	G	T	A	T	A	C	G	T	C
S_2 :				A	T	A	C	A	C	T	T	A	T
P :					1	1	0	0	1	0	1		

as the same spaced word '$TA**C*T$' occurs at positions 5 in S_1 and at position 2 in S_2.

In a previous article, we used spaced-word matches to estimate phylogenetic distances between genomic sequences, by considering at the nucleotides aligned to each other at the *don't care* positions of selected spaced-word matches (Leimeister *et al.*, 2017). To remove spurious random spaced-word matches, we applied a simple *filtering procedure*. Based on the following substitution matrix (Chiaromonte *et al.*, 2002)

	A	C	G	T
A	91	−114	−31	−123
C		100	−125	−31
G			100	−114
T				91

we calculated for each spaced-word match the sum of substitution scores of the nucleotide pairs aligned at the *don't-care* positions, and we removed all spaced-word matches with a score below zero; compare also Brejova *et al.* (2005).

A graphical representation of the spaced-word matches between two sequences shows that this procedure can clearly separate random spaced-word matches from true homologies. If we plot for each possible score value s the number of spaced-word matches with score equal to s, we obtain a bimodal distribution with one peak for random matches and a second peak for true homologies. We call such a plot a *spaced-words histogram*, see Figure 1 for an example. For simulated sequence pairs under a simple model of evolution, and with a sufficient number of *don't-care* positions in the underlying pattern, both peaks are approximately normally distributed. For real-world sequences, the random peak is still normally distributed, but the 'homologous' peak is more complex. Even so, using a suitable cut-off value, one can easily distinguish between random matches and true homologies; for the above matrix, a cut-off of zero works well. More examples for *spaced-words histograms* are given in Leimeister *et al.* (2017).

Herein, we propose to use spaced-word matches to calculate *anchor points* for pairwise alignment of genomic sequences. To distinguish between spaced-word matches representing *true homologies* and random *background* matches, we use the above filtering criterion. More precisely, our approach to find anchor points for genomic alignment is as follows. For given parameters ℓ and w, we first calculate a pattern P with length ℓ and weight w—i.e. with w *match positions*—using our recently developed software *rasbhari* (Hahn *et al.*, 2016). We then identify all spaced-word matches with respect to P. Based on the above substitution matrix, we calculate the *score* of each spaced-word match, and we discard all spaced-word matches with a score below zero, as we did in our previous article (Leimeister *et al.*, 2017). By default, our program uses only *unique* spaced-word matches. That is, if a spaced word w occurs n times in one sequence and m times in a second sequence, we only use the best-scoring of the $n \times m$ resulting spaced-word matches. But as an alternative, it is also possible to use *all* spaced-word matches with a score above zero.

To find homologies even for distantly related sequences, we use patterns with a low weight; by default, we use a weight of $w = 10$. On the other hand, we use a large number of *don't-care* positions, since this makes it easier to distinguish true homologies from random spaced-word matches. By default, we use a pattern length of $\ell = 110$, so our patterns contain 10 match positions and 100 don't-care positions.

Next, we do gap-free extensions of the identified local similarities in both directions using a standard X-drop approach. As starting points for these extensions, we do not use the full spaced-word matches, but their midpoints. The reason for this is that, with our long patterns, even high-scoring spaced-word matches may not represent true homologies over their entire length. It often happens that parts of a spaced-word aligns homologous nucleotides, but one or both ends of the aligned segments extend into non-homologous regions. There is a high probability, however, that the midpoint of a long, high-scoring spaced-word match is located within a region of true homology. As a result, it is possible that an 'extended' match in our approach is shorter than the initial spaced-word match that was used to define the starting point for the X-drop extension. Also, it can happen that a spaced-word match is located within the 'extension' of a previously processed match. Such matches are redundant and are therefore discarded by our algorithm. Finally, we use the extended gap-free alignments as anchor points for alignment.

2.2 Evaluation

To evaluate *FSWM* and to compare it to a state-of-the-art approach to alignment anchoring, we used the *Mugsy* software system. Here, we used the default version of *FSWM* with *unique* matches, i.e. for each distinct spaced word, only the highest-scoring spaced-word match is used. As mentioned above, the original *Mugsy* uses *MUMmer* to find pairwise anchor points. We replaced *MUMmer* in

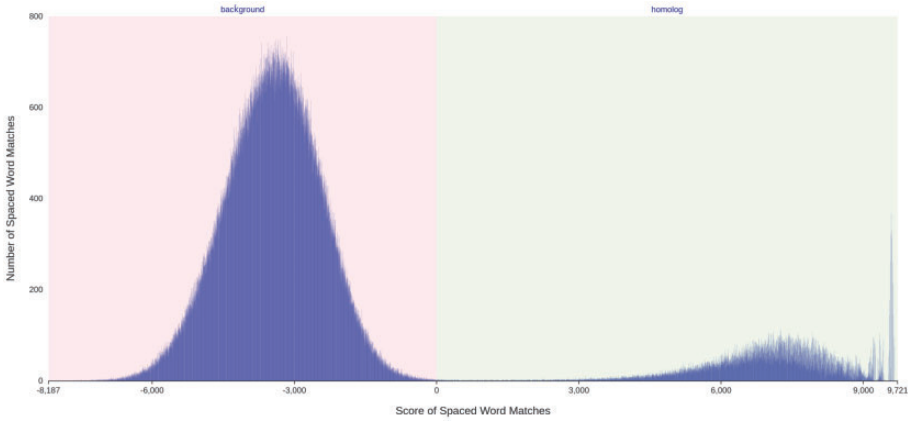

Fig. 1. *Spaced-words histogram* for a comparison of two bacterial genomes, *Phaeobacter gallaeciensis* 2.10 and *Rhodobacterales bacterium Y4I*. All possible spaced-word matches with respect to a given binary pattern P are identified, and their scores are calculated as explained in the main text. The number of spaced-word matches with a score s is plotted against s. Two peaks are visible, an approximately normally distributed peak for background spaced-word matches, and a more complex peak for spaced-word matches representing homologies. With a cut-off value of zero, background and homologous spaced-word matches can be reliably separated

the *Mugsy* pipeline by our *FSWM*-based anchor points and evaluated the resulting multiple alignments. In addition, we compared these alignments to alignments produced by the multiple genome aligner *Cactus* (Paten *et al.*, 2011). *Cactus* is known to be one of the best existing tools for multiple genome alignment; it performed excellently in the *Alignathon* study (Earl *et al.*, 2014). To measure the performance of the compared methods, we used simulated genomic sequences as well as three sets of real genomes. To make *MUMmer* directly comparable to *FSWM*, we used a minimum length of 10 *nt* for maximum unique matches, corresponding to the default *weight* (sum of *match positions*) used in *Spaced Words*. Note that, by default, *MUMmer* uses a minimum length of 15 *nt*. With this default value, however, we obtained alignments of much lower quality. *Cactus* was run with default values.

2.2.1 Simulated genomic sequences

To simulate genomic sequences, we used the *artificial life framework (ALF)* developed by Dalquen *et al.* (2012). *ALF* generates artificial gene families along a randomly generated tree, according to a probabilistic model of evolution. During this process, evolutionary events are logged so the *true* MSA is known for each simulated gene family and can be used as reference to assess the quality of automatically generated alignments.

We generated a series of 14 datasets, each one based on a randomly generated tree with 30 leaves, representing different species. Each dataset consists of 750 simulated gene families, evolved along the respective tree, such that exactly one gene from each family is present in each of the 30 'species'. Within each dataset, we used a fixed mutation rate for all gene families, but we used different mutation rates for different datasets. For all other parameters in *ALF*, we used the default settings. We varied the mutation rates between an *average* of 0.1013 substitutions per position for the first dataset to an average of 0.8349 substitutions per position for the 14th dataset. Here, the average is taken over all pairs of 'species' within the respective dataset. The *maximal* pairwise distance between all pairs of sequences within a dataset ranges from 0.1640 for the first to 1.0923 for the 14th dataset. The simulated genes have an average length of about 1500 *bp*, summing up to a total size of about 32 *MB* per dataset.

For simplicity, we did not concatenate the 750 genes in one 'species'. Instead, we applied the alignment programs that we evaluated to compare all genes from one 'species' to all genes from all other 'species' within the same dataset. Concatenating the sequences would have led to the same results. To assess the quality of the produced alignments, we calculated *recall* and *precision* values in the usual way. If, for one given dataset, S is the set of all positions of the 30×750 simulated gene sequences, we denote by $A \subset \binom{S}{2}$ the set of all pairs of positions aligned to each other by the alignment that is to be evaluated, while $R \subset \binom{S}{2}$ denotes the set of all pairs of positions aligned to each other in the reference alignment. *Recall* and *precision* are then defined as

$$\text{Recall} = \frac{|A \cap R|}{|R|}, \quad \text{Precision} = \frac{|A \cap R|}{|A|} \tag{1}$$

The harmonic mean of *recall* and *precision* is called the *balanced F-score* and is often used as an overall measure of accuracy; it is thus defined as

$$F_{\text{score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

To estimate these three values, we used the tool *mafComparator* which was also used in the *Alignathon* study (Earl *et al.*, 2014). Since it is prohibitive to consider *all* pairs of positions of the test sequences, we sampled 10 million pairs of positions for each dataset. This corresponds to the evaluation procedure used in *Alignathon*.

For the simulated sequence sets, their *recall* and *precision* values are shown in Figures 2 and 3. For datasets with smaller mutation rates, the quality of alignments obtained with *FSWM* and *MUMmer* is comparable (Fig. 4). However, if the mutation rate increases, our spaced-words approach clearly outperforms the original version of *Mugsy* where exact word matches are used to find anchor points. With *FSWM*, not only more homologies are detected, compared to *Mummer*, but also the *precision* of *Mugsy* is slightly improved.

2.2.2 Real-world genome sequences

For real-world genome families, it is usually not possible to calculate the *precision* of MSA programs because it is, in general, not known which sequence positions exactly are homologous to each other and which ones are not. If there are *core blocks* of the sequences for which biologically correct alignments are known, at least *recall* values can be calculated for these core blocks. For most genome

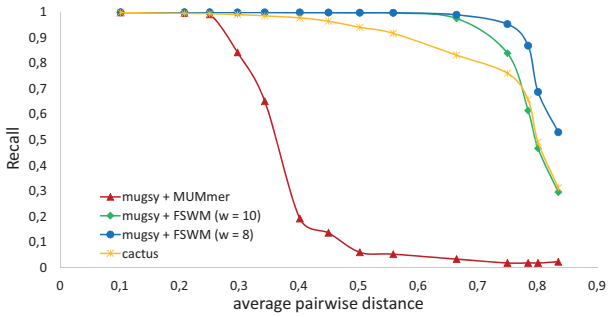

Fig. 2. *Recall* values for *Mugsy* using anchor points generated with *FSWM* and with *MUMmer*, respectively, as well as for *Cactus*. Test data were simulated genomic sequences generated with *ALF*, see main text for details. *FSWM* was run with the default *weight* $w = 10$, i.e. with 10 *match positions* in the underlying pattern, and with $w = 8$

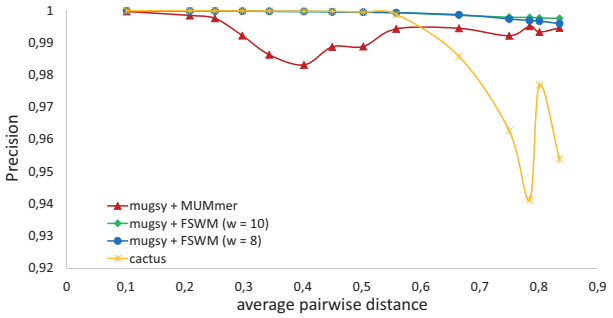

Fig. 3. *Precision* values for *Mugsy* with *FSWM* and *MUMmer* anchor points respectively, and for *Cactus*. Test data and parameter values as in Figure 2

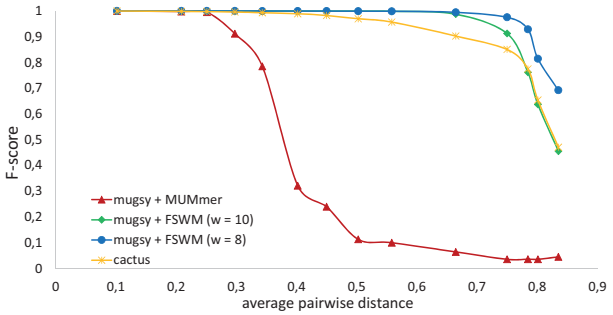

Fig. 4. *F-Score* values for *Mugsy* with *FSWM* and *MUMmer* anchor points, respectively, and for *Cactus*. Test data and parameter values as in Figure 2

sequences, however, not even such core blocks are available. To evaluate *Mugsy*, the authors of the program therefore used the *number of core columns* of the produced alignments as a criterion for alignment quality (Angiuoli and Salzberg, 2011). Here, a *core column* is defined as a column that does not contain gaps, i.e. a column in which nucleotides from *all* of the input sequences are aligned. In addition, the authors of *Mugsy* used the *number of pairs of aligned positions* of the aligned sequences as an indicator of alignment quality. In this article, we use the same criteria to evaluate multiple alignments of real-world genomes.

As a first real-word example, we used a set of 29 *E. coli/Shigella* genomes that has been used in the original *Mugsy* paper, see Supplementary Material for details; these sequences have also been used to evaluate alignment-free methods (Haubold *et al.*, 2015; Morgenstern *et al.*, 2015; Yi and Jin, 2013). The total size of this dataset is about 141 *MB*. As a second test set, we used another prokaryotic dataset, namely a set of 32 complete *Roseobacter* genomes (details in the Supplementary Material); these genomes are more distantly related than the *E. coli/Shigella* strains. The total size of this dataset is about 135 *MB*. To test our approach on eukaryotic genomes, we used as a third test case a set of nine fungal genomes, namely *Coprinopsis cinerea*, *Neurospora crassa*, *Aspergillus terreus*, *Aspergillus nidulans*, *Histoplasma capsulatum*, *Paracoccidioides brasiliensis*, *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe* and *Ustilago maydis* (genbank accession numbers are given in the Supplementary Material). The total size of this third dataset is about 253 *MB*.

Table 1. Evaluation of multiple alignments of 29 *E. coli/Shigella* genomes, 32 *Roseobacter* genomes and 9 fungal genomes, obtained with *Mugsy*, using anchor points calculated with *FSWM* and with *MUMmer*, respectively

	Core LCBs	Aligned pairs	Core col.	LCBs
29 *E. coli/Shigella* genomes				
Mugsy + MUMmer	539	1,61E+09	2,827,115	4138
Mugsy + FSWM	664	1,63E+09	2,867,432	5906
Cactus	20,163	1,48E+09	2,663,750	56,592
32 *Roseobacter* genomes				
Mugsy + MUMmer	39	3,63E+08	13,654	13,501
Mugsy + FSWM	859	7,15E+08	824,054	30,836
Cactus	5984	4,95E+08	280,085	337,320
9 fungal genomes				
Mugsy + MUMmer	9	5,88E+06	2097	4252
Mugsy + FSWM	2590	1,18E+08	718,176	89,555
Cactus	31,589	1,33E+08	828,680	848,242

Note: As a comparison, the table contains the results obtained with *Cactus*. The first column contains the number of *core columns*, i.e. the number of columns in the multiple alignments that do not contain gaps; the second column contains the total number of aligned pairs of positions in the alignment. The third column contains the number of *core Locally Collinear Blocks (LCBs)* i.e. the number of *LCBs* that involve *all* of the aligned genomes ('core LCBs'), while the last column contains the total number of *LCBs*.

The results of *Mugsy* with *MUMmer* and *FSWM*, respectively, for the three real-world datasets are shown in Table 1, together with the results obtained with *Cactus*. In addition to the number of *core columns* and the number of aligned pairs of positions, the table contains the number of *core Locally Collinear Blocks*, i.e. the number of *Locally Collinear Blocks* involving all of the input sequences, and the total number of *Locally Collinear Blocks* returned by the alignment programs. For the *E. coli/Shigella* sequences, the two anchoring methods, *MUMmer* and *FSWM*, led to alignments of comparable quality when used with *Mugsy*; the genome sequences in this dataset are very similar to each other. For the *Roseobacter* and fungal genomes, however, the *FSWM* anchor points led to much better alignments than the default anchor points generated with *MUMmer*. The sequences in these sets are far more apart from each other than the sequences in the *E. coli/Shigella* set, so the results on these three datasets confirm our above results on simulated sequences.

2.2.3 Program run time

Table 2 reports the program run times of *Mugsy* with *FSWM*, *Mugsy* with *MUMmer* and *Cactus* on the above three real-world sequence sets. In addition, the table contains the run times for *FSWM* and *MUMmer* alone. A program run of *Mugsy* with *FSWM* on a set

Table 2. Run time in minutes for three different multiple genome-alignment methods applied to the three test datasets that we used in our program evaluation

	E. coli/Shigella	*Roseobacter*	*fungal genomes*
FSWM	59	83	110
FSWM + Mugsy	638	6428	1488
MUMmer	73	63	43
MUMmer + Mugsy	286	1099	63
Cactus	714	1775	775

of five mammalian sequences of length 200 *mb* each from Earl *et al.* (2014) took around 7 days, and 5 h with $k = 10$ and two days with $k = 12$.

3 Discussion

In this article, we proposed a novel approach to calculate anchor points for genome alignment. Finding suitable anchor points is a critical step in all methods for genome alignment, since the selected anchor points determine which regions of the sequences can be aligned to each other in the final alignment. A sufficient number of anchor points is necessary to keep the search space and run time of the main alignment procedure manageable, so *sensitive* methods are needed to find anchor points. Wrongly selected anchor points, on the other hand, can seriously deteriorate the quality of the final alignments, so anchoring procedures must also be highly *specific*.

Earlier approaches to genomic alignment used exact word matches as anchor points (Delcher *et al.*, 1999; Höhl *et al.*, 2002), since such matches can be easily found using suffix trees and related indexing structures. These approaches are limited, however, to situations where closely related genomes are to be aligned, for example different strains of a bacterium. In modern approaches to database searching, *spaced seeds* are used to find potential sequence homologies (Buchfink *et al.*, 2015; Hauswedell *et al.*, 2014; Li *et al.*, 2003). Here, binary patterns of *match* and *don't care* positions are used, and two sequence segments of the corresponding length are considered to match if identical residues are aligned at the *match* positions, while mismatches are allowed at the *don't care* positions. Such pattern-based approaches are more *sensitive* than previous methods that relied on exact word matches.

We previously proposed to apply the 'spaced-seeds' idea to alignment-free sequence comparison, by replacing contiguous words by so-called *spaced words*, i.e. by words that contain wildcard characters at certain pre-defined positions (Leimeister *et al.*, 2014). More recently, we introduced *FSWM* (Leimeister *et al.*, 2017) to estimate the average number of substitutions per sequence position between two genomes. In the latter approach, we first identify spaced-word matches using relatively long patterns with only few *match* positions. For the identified matching segments, we look at the nucleotides that are aligned to each other at the *don't-care* positions, and we discard spaced-word matches for which the similarity at the *don't-care* positions is below a threshold. Substitution frequencies are then estimated based on the aligned nucleotides at the don't-care positions of the remaining spaced-word matches. We showed that this procedure is fast and highly sensitive, and it can reliably distinguish between true homologies and spurious sequence similarities.

In the present study, we used FSWM to calculate anchor points for genomic sequence alignment. Instead of using the selected spaced-word matches directly as anchor points, we extend the identified hits into both directions, similar to the *hit-and-extend* approach to database searching. In view of speed and accuracy, this approach is somewhere between exact word matching and gapped local alignment. As in our previous paper on filtered spaced words (Leimeister *et al.*, 2017), we use binary patterns with a large number of *don't-care* positions. This way, the 'homologous' and 'background' peaks in the *spaced-word histograms* (Fig. 1) are far enough apart, since the distance between them is proportional to the number of *don't-care* positions in the underlying patterns. With a large number of *don't-care positions*, it is therefore easier to distinguish between homologous and background spaced-word matches.

One might think that, with our long patterns, we might miss too many shorter local homologies. We do not see this as a problem, though. Our goal is not to find *all* local homologies between two sequences, but to output a sufficient number of anchor points to make the final alignment procedure feasible. Moreover, our algorithm is well able to find gap-free homologies that are shorter than the specified pattern length, as long as the sequence similarity between these homologies is strong enough. As explained above, we do not start the *X*-drop extension at the end positions of the identified hits, but in the middle; this way we can find spaced-word matches that cover short homologies, but reach into gapped or non-homologous sequence regions to the left and to the right. In such cases, it can happen that the 'extended' hits are *shorter* than the respective initial spaced-word matches.

To evaluate these anchor points, we integrated them into the popular genome-alignment pipeline *Mugsy*. Test runs on simulated genome sequences show that, for closely related sequences, *Mugsy* produces alignments of high quality with both types of anchor points. For more distantly related sequences, however, the *recall* values of the program drop dramatically if anchor points are calculated with *MUMmer* while, with our spaced-word matches, one observes recall values close to 100% for distances up to around 0.7 substitutions per position.

For real-world genomes, it is more difficult to evaluate the performance of genome aligners since there is only limited information available on which positions are homologous to each other and which ones are not. Angiuoli and Salzberg (2011) therefore used the number of aligned pairs of positions as an indicator of alignment quality, together with the size of the 'core alignment', i.e. the number of alignments columns that do not contain gaps. At first glance, these criteria might seem questionable; it would be trivial to maximize these values, simply by aligning sequences without internal gaps, by adding gaps only at the ends of the shorter sequences. However, as shown in Figure 3, all MSA programs in our study have high *precision* values, i.e. positions aligned by these programs are likely to be true homologs. In this situation, the number of aligned position pairs and size of the 'core alignment' can be considered as a proxy for the *recall* of the applied methods i.e. the proportion of homologies that are correctly aligned.

As shown in Table 2, the program run time to generate anchor points is comparable for *FSWM* and *MUMmer*. For distantly related sequence sets, however, the *total* run time of *Mugsy* is much higher with our *FSWM* anchoring approach than with anchor points from *MUMmer*. A possible explanation for the difference in run time is that *FSWM* is more sensitive, so a larger number of anchor points are produced. Table 1 shows that, with our FSWM, more *Locally Collinear Blocks* are found than with the exact word matches that are found with *MUMmer*—especially for distantly related sequences where exact word matching is not very sensitive. One way of reducing the program run time would be to apply a cut-off value to reduce the number *Locally Collinear Blocks* that are to be aligned in the main alignment procedure. Further research efforts are necessary to

balance speed and accuracy of multiple genome alignment algorithms.

Acknowledgements

We thank Rasmus Steinkamp for IT support. Three anonymous reviewers read the manuscript very carefully and made many detailed and helpful comments.

Funding

The project was partially funded by the VW Foundation, project VWZN3157. We acknowledge support by the *Open Access Publication Funds* of the *Göttingen University*.

Conflict of Interest: none declared.

References

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Angiuoli,S.V. and Salzberg,S.L. (2011) Mugsy: fast multiple alignment of closely related whole genomes. *Bioinformatics*, **27**, 334–342.

Batzoglou,S. (2005) The many faces of sequence alignment. *Brief. Bioinformatics*, **6**, 6–22.

Blanchette,M. *et al.* (2004) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, **14**, 708–715.

Bradley,R.K. *et al.* (2009) Fast statistical alignment. *PLoS Comput. Biol.*, **5**, e1000392.

Bray,N. and Pachter,L. (2003) MAVID multiple alignment server. *Nucleic Acids Res.*, **31**, 3525–3526.

Bray,N. *et al.* (2003) AVID: a Global Alignment Program. *Genome Res.*, **13**, 97–102.

Brejova,B. *et al.* (2005) Vector seeds: an extension to spaced seeds. *J. Comp. Syst. Sci.*, **70**, 364–380.

Břinda,K. *et al.* (2015) Spaced seeds improve *k*-mer-based metagenomic classification. *Bioinformatics*, **31**, 3584–3592.

Brudno,M. *et al.* (2003a) Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinformatics*, **4**, 66.

Brudno,M. *et al.* (2003b) LAGAN and multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res.*, **13**, 721–731.

Buchfink,B. *et al.* (2015) Fast and sensitive protein alignment using DIAMOND. *Nat. Methods*, **12**, 59–60.

Chiaromonte,F. *et al.* (2002) Scoring pairwise genomic sequence alignments. In: Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E. (eds.) *Pacific Symposium on Biocomputing*. Lihue, Hawaii, pp. 115–126.

Choi,K.P. *et al.* (2004) Good spaced seeds for homology search. *Bioinformatics*, **20**, 1053.

Dalquen,D.A. *et al.* (2012) ALF: a simulation framework for genome evolution. *Mol. Biol. Evol.*, **29**, 1115–1123.

Darling,A.C.E. *et al.* (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res.*, **14**, 1394–1403.

Darling,A.E. *et al.* (2006) Procrastination leads to efficient filtration for local multiple alignment. In: Bucher, P. and Moret, B. M. E. (eds.) *Algorithms in Bioinformatics, Lecture Notes in Bioinformatics*. Springer, Berlin, Germany, pp. 126–137.

Darling,A.E. *et al.* (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS One*, **5**, e11147.

David,M. *et al.* (2011) SHRiMP2: sensitive yet practical short read mapping. *Bioinformatics*, **27**, 1011–1012.

Delcher,A.L. *et al.* (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.

Deneke,C. *et al.* (2017) PaPrBaG: a machine learning approach for the detection of novel pathogens from NGS data. *Sci. Rep.*, **7**, 39194.

Dewey,C.N. and Pachter,L. (2006) Evolution at the nucleotide level: the problem of multiple whole-genome alignment. *Hum. Mol. Genet.*, **15**, R51–R56.

Döring,A. *et al.* (2008) SeqAn—an efficient, generic C++ library for sequence analysis. *BMC Bioinformatics*, **9**, 11.

Dubchak,I. *et al.* (2009) Multiple whole-genome alignments without a reference organism. *Genome Res.*, **19**, 682–689.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis*. Cambridge University Press, Cambridge, UK.

Earl,D. *et al.* (2014) Alignathon: a competitive assessment of whole-genome alignment methods. *Genome Res.*, **24**, 2077–2089.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Hahn,L. *et al.* (2016) *rasbhari*: optimizing spaced seeds for database searching, read mapping and alignment-free sequence comparison. *PLoS Comput. Biol.*, **12**, e1005107.

Haubold,B. *et al.* (2015) andi: fast and accurate estimation of evolutionary distances between closely related genomes. *Bioinformatics*, **31**, 1169–1175.

Hauswedell,H. *et al.* (2014) Lambda: the local aligner for massive biological data. *Bioinformatics*, **30**, i349–i355.

Höhl,M. *et al.* (2002) Efficient multiple genome alignment. *Bioinformatics*, **18**, S312–320S.

Horwege,S. *et al.* (2014) *Spaced words* and *kmacs*: fast alignment-free sequence comparison based on inexact word matches. *Nucleic Acids Res.*, **42**, W7–W11.

Huang,W. *et al.* (2006) Accurate anchoring alignment of divergent sequences. *Bioinformatics*, **22**, 29–34.

Kurtz,S. (1999) Reducing the space requirement of suffix trees. *Softw. Pract. Exp.*, **29**, 1149–1171.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12. +.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Leimeister,C.-A. *et al.* (2014) Fast Alignment-Free sequence comparison using spaced-word frequencies. *Bioinformatics*, **30**, 1991–1999.

Leimeister,C.-A. *et al.* (2017) Fast and accurate phylogeny reconstruction using filtered spaced-word matches. *Bioinformatics*, **33**, 971–979.

Li,M. *et al.* (2003) PatternHunter II: highly sensitive and fast homology search. *Genome Informatics*, **14**, 164–175.

Li,M. *et al.* (2006) Superiority and complexity of the spaced seeds. In: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, SODA '06*, pp. 444–453. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Ma,B. *et al.* (2002) PatternHunter: faster and more sensitive homology search. *Bioinformatics*, **18**, 440–445.

Morgenstern,B. (2002) A simple and space-efficient fragment-chaining algorithm for alignment of DNA and protein sequences. *Appl. Math. Lett.*, **15**, 11–16.

Morgenstern,B. *et al.* (1996) Multiple DNA and protein sequence alignment based on segment-to-segment comparison. *Proc. Natl. Acad. Sci. USA*, **93**, 12098–12103.

Morgenstern,B. *et al.* (2002) Exon discovery by genomic sequence alignment. *Bioinformatics*, **18**, 777–787.

Morgenstern,B. *et al.* (2006) Multiple sequence alignment with user-defined anchor points. *Algorith. Mol. Biol.*, **1**, 6.

Morgenstern,B. *et al.* (2015) Estimating evolutionary distances between genomic sequences from spaced-word matches. *Algorith. Mol. Biol.*, **10**, 5.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Noé,L. (2017) Best hits of 11110110111: model-free selection and parameter-free sensitivity calculation of spaced seeds. *Algorith. Mole. Biol.*, **12**.

Noé,L. *et al.* (2010) Designing efficient spaced seeds for SOLiD read mapping. *Adv. Bioinformatics*, **2010**, 1–12.

Ogurtsov,A.Y. *et al*. (2002) OWEN: aligning long collinear regions of genomes. *Bioinformatics*, **18**, 1703–1704.

Ounit,R. and Lonardi,S. (2015) Higher classification accuracy of short metagenomic reads by discriminative spaced k-mers. In: *Algorithms in Bioinformatics: 15th International Workshop, WABI 2015, Atlanta, GA, USA, September 10–12, 2015, Proceedings*, pp. 286–295. Springer, Berlin, Germany.

Paten,B. *et al*. (2011) Cactus: algorithms for genome multiple sequence alignment. *Genome Res*., **21**, 1512–1528.

Raphael,B. *et al*. (2004) A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res*., **14**, 2336–2346.

Rausch,T. *et al*. (2008) Segment-based multiple sequence alignment. *Bioinformatics*, **24**, i187–i192.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

The Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature*, **408**, 796–815.

Xu,J. *et al*. (2006) Optimizing multiple spaced seeds for homology search. *J. Comput. Biol*., **13**, 1355–1368.

Yi,H. and Jin,L. (2013) Co-phylog: an assembly-free phylogenomic approach for closely related organisms. *Nucleic Acids Res*., **41**, e75.

4 *Prot-SpaM*: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences

Reference

Chris-Andre Leimeister, Jendrik Schellhorn, Svenja Schöbel, Michael Gerth, Christoph Bleidorn, and Burkhard Morgenstern. Prot-SpaM: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *GigaScience*, 2018 to appear. [63]

Original Contribution

CAL designed the study, co-supervised the project and drafted parts of the manuscript. JS implemented the software and did most of the program evaluation. SS contributed to the program evaluation. MG and CB analysed and discussed the results on Wolbachia and wrote parts of the manuscript. BM co-supervised the project and wrote most of the manuscript. All authors read and approved the final version of the manuscript.

Prot-SpaM: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences

Chris-Andre Leimeister[1,*], Jendrik Schellhorn [1,*], Svenja Dörrer[1], Michael Gerth [2], Christoph Bleidorn [3,4], and Burkhard Morgenstern [1,5]

[1]University of Göttingen, Department of Bioinformatics, Goldschmidtstr. 1, 37077 Göttingen, Germany
[2]Institute for Integrative Biology, University of Liverpool, Biosciences Building, Crown Street, L69 7ZB Liverpool, UK
[3]University of Göttingen, Department of Animal Evolution and Biodiversity, Untere Karspüle 2, 37073 Göttingen, Germany
[4]Museo Nacional de Ciencias Naturales, Spanish National Research Council (CSIC), 28006 Madrid, Spain
[5]Göttingen Center of Molecular Biosciences (GZMB), Justus-von-Liebig-Weg 11, 37077 Göttingen
[*]Joint first authors

October 30, 2018

Abstract

Word-based or 'alignment-free' sequence comparison has become an active research area in bioinformatics. While previous word-frequency approaches calculated rough measures of sequence similarity or dissimilarity, some new alignment-free methods are able to accurately estimate phylogenetic distances between genomic sequences. One of these approaches is *Filtered Spaced Word Matches*. Herein, we extend this approach to estimate evolutionary distances between complete or incomplete proteomes; our implementation of this approach is called *Prot-SpaM*. We compare the performance of *Prot-SpaM* to other alignment-free methods on simulated sequences and on various groups of eukaryotic and prokaryotic taxa. *Prot-SpaM* can be used to calculate high-quality phylogenetic trees for dozens of whole-proteome sequences in a matter of seconds or minutes and often outperforms

other alignment-free approaches. The source code of our software is available through *Github*: `https://github.com/jschellh/ProtSpaM`

1 Introduction

Evolutionary relationships between species are usually inferred by comparing homologous gene or protein sequences to each other. Here, groups of orthologous sequences have to be identified first, for which then multiple alignments are to be calculated. There are generally two different strategies of resolving phylogenies based on multiple alignments. In the so-called *supermatrix* approach, multiple sequence alignments of single genes or proteins are concatenated. A phylogenetic tree is inferred from the resulting matrix, *e.g.*, using *Maximum Likelihood* [63] or *Bayesian inference* [57]. Alternatively, gene or protein trees are inferred for every single multiple sequence alignment and the resulting phylogeny is inferred using *coalescent* models [46] or supertree [4] approaches.

All these steps are time consuming, and often manual intervention is required. Therefore, *word-based* or *alignment-free* alternatives have been proposed recently, which are much faster and which require much less data preparation. Most alignment-free methods compare the *word composition* of sequences [11, 20, 31, 59, 65, 69], with some approaches also considering background word frequencies [53, 54, 60, 70], see [55] for a review of these latter approaches. More recently, the *spaced-word* composition of sequences has been used for sequence comparison [32, 42, 48, 50]. Other alignment-free methods are based on the so-called *matching statistics*, that is they use the length of maximal common subwords [12, 68]. This has been extended to maximal common subwords with a certain number of mismatches [43, 52, 66, 67]. Alignment free approaches have been recently reviewed in detail [2, 27, 75].

Accurate alignment-free tools are urgently needed because of the huge volume of data generated by new sequencing techniques. Another advantage of alignment-free methods, compared to alignment-based approaches, is the fact that they can be applied to incomplete data, for example to unassembled sequencing reads or to partially sequenced genomes [18]. Note that some of the so-called 'alignment-free' approaches are based on comparing words of the input sequences to each other. So, strictly spoken, they are not 'alignment-free' since they align these words to each other. The term *alignment-free* is used nevertheless by most researchers, since these word-based approaches circumvent the need to calculate full pairwise or multiple

alignments of the sequences under study.

The above mentioned approaches to alignment-free sequence comparison calculate ad-hoc measures of sequence similarity or dissimilarity. They are not based on stochastic models of molecular evolution, and they do not try to estimate distances between sequences in a statistically rigorous way. More recently, some alignment-free approaches have been proposed that are based on explicit models of DNA evolution. These methods are able to estimate the number of substitutions per site that have happened since two nucleic-acid sequences have evolved from their last common ancestor [14,28,29,44,47,72].

A main application of alignment-free approaches is comparison of whole *genomes*. Consequently, most alignment-free methods have been designed to work on DNA sequences. If distantly related species are studied, though, phylogenetic trees are usually inferred from protein sequences rather than from DNA sequences. The reason for this is that protein sequences are more conserved than DNA sequences, as synonymous substitutions are not visible in proteins. Thus, for distal species, it may be hard to detect similarities between genes at the DNA-sequence level, while homologies may be still detectable among protein sequences. It is therefore highly desirable to have accurate alignment-free software tools that work on protein sequences, in addition to the available tools for DNA sequence comparison. Generic word-frequency methods can be applied to both DNA and protein sequences; the program *FFP*, for example, has been used to whole-proteome comparison [35]. As mentioned above, however, these methods do not estimate phylogenetic distances in a rigorous way. So far, there are no alignment-free approaches available that can accurately estimate evolutionary distances between protein sequences.

In this paper, we propose an alignment-free method which estimates the number of substitutions protein sequences since they evolved from their last common ancestor. Our approach is based on *Filtered Spaced Word Matches (FSWM)*, a concept we introduced recently for whole-genome sequence comparison [44], see [28,72] for related approaches. We call the implementation of this new approach *Prot*eome-based *Spa*ced-Word *M*atches *(Prot-SpaM)*. The basic idea is to use gap-free pairwise alignments of fixed-length words with matching amino-acid residues at certain pre-defined positions. Such *spaced-word matches* can be rapidly identified and, after discarding random background matches, the remaining 'homologous' spaced-word matches can be used to estimate the phylogenetic distance between two taxa. To our knowledge, this is the first approach that accurately estimates evolutionary distances between protein sequences without the need to calculate full sequence alignments.

To evaluate our approach, we used simulated protein sequences and real-world whole proteomes. Test runs on the simulated sequences show that our distance estimates are very close to the true distances, for distance values of up to around 2.0 substitutions per sequence position. On the real-world sequences, we evaluated our approach indirectly, by phylogenetic analysis, as is common practice in the field. We used *Prot-SpaM* to estimate pairwise distances for various sets of taxa, and we applied the *Neighbor-Joining* algorithm [58] to calculate phylogenetic trees from the resulting distance matrices. These trees were finally evaluated by comparing them to reference trees that were determined by standard methods and can be considered to be reliable. We show that the trees obtained with our approach are often of high quality, and they are generally more similar to the respective reference trees than trees generated with other alignment-free approaches.

2 Method

We consider sequences over an alphabet $\mathcal{A}$. In this paper, $\mathcal{A}$ consists of 20 characters representing the 20 different amino acids. *Prot-SpaM* is based on so-called *spaced-word matches* between sequences. For a *wildcard character* '$*$' with $* \notin \mathcal{A}$ and a binary pattern P of length ℓ – *i.e.* for a length-ℓ word P over $\{0,1\}$ –, a *spaced-word* with respect to P is a length-ℓ word W over the alphabet $\mathcal{A} \cup \{*\}$ such that $W(i) = *$ if and only if $P(i) = 0$. An index $i \in \{1, \ldots, \ell\}$ is called a *match position* of P or W, respectively, if $P(i) = 1$, and a *don't care position* otherwise. The number of *match* positions in a pattern or spaced-word is called its *weight* w. We say that a spaced word W with respect to P occurs in a sequence S at some position i if one has $W(k) = S(i + k - 1)$ for all $k \in \{1, \ldots, \ell\}$ with $P(k) = 1$ – *i.e.* for all *match positions* of P.

Moreover, we say that there is a *spaced-word match* w.r.t. P between two sequences S_1 and S_2 at (i_1, i_2) if the same spaced word w.r.t P occurs at position i_1 in S_1 and at position i_2 at S_2. In other words, there is a spaced-word match between S_1 and S_2 at (i_1, i_2), if and only if one has $S_1(i_1 + k - 1) = S_2(i_2 + k - 1)$ for all match positions k of P. Below is an example for a spaced-word match between two sequences S_1 and S_2 at $(2, 3)$ with respect to the pattern $P = 1100101$; the spaced word $TN**D*P$ occurs at position 2 in S_1 and at position 3 in S_2:

S_1 :		T	T	N	Q	I	D	L	P	P	C	Y	N
S_2 :	A	C	T	N	L	I	D	I	P	Q	N		
P :			1	1	0	0	1	0	1				

Similar to our original *FSWM* approach, we estimate distances between protein sequences based on selected spaced-word matches between them, with respect to one or several pre-defined patterns. Distance values are obtained by comparing the amino-acid residues that are aligned to each other at the *don't-care* positions of the selected spaced-word matches. This is similar to estimating distances in standard alignment-based approaches – the only difference to those standard approaches is that we are using *don't-care positions* of spaced-word matches instead of full sequence alignments.

To estimate distances in this way, one has to make sure that only those spaced-word matches are selected that represent *homologies*, *i.e.* that the involved spaced-word occurrences go back to the same origin in the last common ancestor of the two proteins that are compared. To distinguish such 'homologous' spaced-word matches from random background matches, we calculate a *score* for each spaced-word match using the BLOSUM62 substitution matrix [30]. Similar to the previous version of our program for nucleic-acid sequences, we define the *score* of a spaced-word match as the sum of substitution scores of the aligned amino acids at the *don't-care* positions. Based on this score, our algorithm decides if a spaced-word match is homologous or not: if its score is below a certain threshold T, then a spaced-word match is considered a random match and is not further considered. As default we use a threshold value of $T = 0$. To see that this threshold accurately separates homologous from background spaced-word-matches, one can plot the *number* of spaced-word matches with a score s against s, see Figure 1; we call such a plot a *Spaced-word-Match histogram* or *spamogram*, for short. In these plots, two peaks are typically visible, a peak on the right-hand side representing *homologous* spaced-word matches and a peak on the left-hand side representing *background* matches. By default, we are using patterns with a weight of $w = 6$ and with 40 *don't-care* positions, *i.e.* with a length of $\ell = 46$.

Moreover, we use a one-to-one mapping of spaced-word occurrences. Note that, if sequences S_1 and S_2 are compared and a spaced word W occurs n time in S_1 and n' times in S_2, than this gives rise to a total of $n \times n'$ spaced-word matches. Taking all these spaced-word matches into account for phylogeny reconstruction, would over-emphasize repeated regions where the same spaced words occur multiple times. Instead of using all possible spaced-word matches, we therefore use a one-to-one mapping of spaced-

word occurrences in the compared sequences. That is, we ensure, that each spaced word occurrence is involved in most one of the selected spaced-word matches. Formally, if there are two spaced word matches, at (i_1, i_2) and at (j_1, j_2), respectively, then we can include both of them simultaneously in our list of selected spaced-word matches, only if $i_1 \neq j_1$ and $i_2 \neq j_2$ hold. To achieve this, we use the same *greedy* algorithm that we described in our previous paper [44]: for a given spaced word W, we calculate the scores of all spaced-word matches involving W. We then select them one-by-one in descending order of their scores – always ensuring that each *occurrence* of W is used in at most one of the selected spaced-word matches.

Finally, in order to estimate pairwise distances between two input sequences, we consider the pairs of amino acids aligned to each other at the don't-care positions of the selected spaced-word matches. Here, we are using the *Kimura* model [38] that approximates the *PAM* distance [13] between sequences based on the number of mismatches per position. We are using these two different models since the *Kimura* model is commonly used to infer distances from the number of mismatches per position in alignments. The *BLOSUM* matrices, on the other hand, are standard in homology searching. Generally, our procedure to filter out background spaced-word matches is rather robust since the homologous and background regions in our *spamograms* can be easily distinguished as can be seen, for example, in Figure 1. So the choice of the substitution matrix to distinguish homologous from background spaced-word matches does not affect the results of our approach too much.

The accuracy and statistical stability of the described approach depends on the number of selected spaced-word matches: the more matches we obtain, the more accurate and stable the results of our method will be. To increase the number of spaced-word matches, the default version of our program uses *multiple* patterns, instead of one single pattern P. More precisely, we are using a set $\mathcal{P} = \{P_1, \ldots, P_m\}$ of m binary patterns, such that all patterns in $\mathcal{P}$ have the same length ℓ and the same weight w, but have their *match* and *don't-care* positions arranged differently; we then use spaced-word matches with respect to *all* patterns $P_i \in \mathcal{P}$. By default, our program uses sets of $m = 5$ patterns. To find suitable patterns sets, we integrated the tool *rasbhari* [25] into our implementation. *rasbhari* uses a *hill climbing* algorithm to optimize pattern sets according to a user-defined criterion. In our program, we use *rasbhari* to minimize the *overlap complexity* [33] of pattern sets. Note that *rasbhari* uses a probabilistic algorithm. It is therefore possible that different program runs of *rasbhari* return different pattern sets, even if the same parameter values are used. Consequently, different runs of

Prot-SpaM on the same sequences and with the same parameter setting can produce slightly different distance estimates.

3 Results

To assess the quality of our new approach and to compare it to other alignment-free methods, we used artificially generated as well as real-world protein sequences. For the test runs we used the default parameters of our program, namely 6 match positions and 40 don't care positions – *i.e.* a total pattern length of 46 –, a threshold of $T = 0$ to discard background spaced-word matches, and sets of $m = 5$ patterns. We compared our program to four other alignment-free methods that can be run on protein sequences, namely *ACS* [68], *FFP* [35, 59], *kmacs* [43] and *CVTree* [53]. Here, we used version 3.19 of *FFP*, the other programs that we evaluated did not have version numbers at the time of writing. Since the original implementation of *ACS* is not publicly available, we used our own implementation of this approach by running *kmacs* with $k = 0$. The competing tools, too, were used with their default parameters. In addition to evaluating these tools on protein sequences, we ran *Filtered Spaced Word Matches* on the complete *genome* sequences of the same taxa. All test runs were done on a 10 x *Intel(R) Xeon(R)* CPU E7-4850 with 2.00 *GHz* with 4 cores each summing up to 40 cores and 1000 *GB RAM*.

3.1 Distance Estimation on Simulated Sequences

To evaluate the distances estimated by our program, we simulated sequences with the tool *pyvolve* [61]. *Pyvolve* simulates sequences along an evolutionary tree using continuous-time Markov models. It can use various substitution models such as *JTT* [34] and other models. Since there are no reliable stochastic models for insertions and deletions in protein sequences, the program produces indel-free sequences. We simulated pairs of sequences of length 100,000 with distances between 0 and 2 substitutions per position, in steps of 0.05, using the *JTT* model. To evaluate the estimated distance values, we generated 1,000 sequence pairs for each distance value and plotted the average of the estimated distances against the real *Kimura* distance of the respective sequence pairs, calculated with the program *protdist* from the *phylip* package [19]. To study the robustness of the estimated distances, we added error bars representing standard deviations to the plot. In addition to running *Prot-SpaM* with default parameters – *i.e.* with sets $\mathcal{P}$ of $m = 5$

patterns –, we did a second series of test runs with $m = 1$, *i.e.* with single patterns. Figure 2 shows the results of these test runs.

3.2 Phylogenetic tree reconstruction

Next, we applied the above alignment-free methods to calculate phylogenetic trees from real-world protein sequences. For four different groups of species, we downloaded all available protein sequences from *GenBank* [1]; within each group, we calculated all pairwise distances between the species. We used the distance matrices obtained in this way as input for *Neighbor-Joining* [58] and compared the resulting trees to reference trees which we assume to reflect the respective correct phylogeny for each group. The *Robinson-Foulds (RF)* distances [56] between the reconstructed trees and the respective reference trees are shown in Table 2.

As mentioned above, *Prot-SpaM* uses a probabilistic algorithm to generate pattern sets, so the results of different program runs on a sequence set can slightly differ, even if the same parameter values are used. We therefore performed 100 program runs on each data set, and Table 2 reports the *average RF* distances for these 100 program runs. An exception was the large prokaryotic data set where we only performed one single program run. Since *absolute RF* distances are not easy to interpret, Table 2 also reports the *relative RF* distances, which are obtained from the *absolute RF* distances by dividing by the maximum possible *RF* distance for a given data set. The maximal possible *RF* distances for a set of n taxa is $2 \cdot n - 6$ [9]. Program run times for the different approaches are shown in Table 3. Trees were visualized with *iTOL* [45]. *Neighbor-Joining* trees and *Robinson-Foulds* distances were calculated with the *phylip* package [19].

E. coli / Shigella

Our first data set consists of 29 strains of *Escherichia coli* and *Shigella*. For each strain, we were able to download about 4,000-5,000 protein sequences; the total size of this data set is around 41 *MB*. Figure 5 shows the reference tree that we used and the tree obtained with the algorithm described in this paper. The reference tree was published by Zhou *et al.* [74] and is based on a multiple sequence alignment of 2,034 core genes and a *Maximum Likelihood* method. As can be seen in Table 2, our approach produced a tree with a topology almost identical to the reference tree. All of the 100 program runs that we performed with *Prot-SpaM* produced the same tree topology; the *RF* distance between these trees and the reference tree was 4.

The other protein-based alignment-free methods led to phylogenies with *RF* distances to the reference tree between 24 and 42, while the genome-based tree obtained with *FSWM* had a *RF* distance of 6 to the reference tree. These trees are shown in the supplementary material.

Wolbachia

As a second test case for benchmarking, we analysed the phylogeny of *Wolbachia* strains, a group of Alphaproteobacteria which are intracellular endosymbionts of arthropods and nematodes [71]. Within *Wolbachia*, 16 distinct genetic lineages (supergroups) are currently distinguished (named by capital letters A-F and H-Q), which may differ in host specificity and type of symbiosis [24]. We re-analyzed a phylogenomic data set by [22], thereby focusing on relationships of strains within supergroups (*Wolbachia I*). A tree generated with *Prot-SpaM* from this data set is shown Figure 4.

For a second *Wolbachia* benchmarking data set, we analysed relationships between supergroups based on available (draft) genomes, see below (*Wolbachia II*). For within supergroup relationships (*Wolbachia I*), a program run of *Prot-SpaM* on the whole proteome recovered a tree which is largely congruent in topology and branch lengths in comparison to a phylogenomic supermatrix analysis of 252 single-copy orthologs which excluded genes which showed signs of recombination. A comparison based on *RF* distances showed that our new method out-competes other available alignment-free programs (Table 2). Interestingly, when only analysing the 252 ortholog data set of [22] instead of whole proteomes, *RF* distances become bigger, and other alignment-free method perform better (Table 2).

Analysing relationships between supergroups has been historically regarded as difficult phylogenetic problem [5, 23]. Analysing all annotated proteins from available genomes with *Prot-SpaM* supported the monophyly of all supergroups. Moreover, this analysis found the same *Wolbachia* strains basally branching as recent analyses suggested. Surprisingly, the phylogenomic supermatrix analysis analysis of 252 single-copy orthologs which excluded genes which showed signs of recombination of this data set recovered a topology which differs to previous study in not supporting the sister group relationship of supergroups A and B. In contrast, as found in previous analyses, the sister group relationship of supergroups A and B is supported by the *Prot-SpaM* analysis. The *Prot-SpaM* analysis also recovered some relationships between supergroups which differ from the topologies of our phylogenomic analysis or expectations from a recently published phylogenomic study [7]. However, it is known that supergroups differ in their base

(and amino acid) composition, and it is currently unknown how this may impact alignment free methods. More sophisticated evolutionary models could alleviate these differences in future studies. Nevertheless, in this test case *Prot-SpaM* also outperforms other alignment free methods when comparing the resulting phylogenetic tree with a phylogenomic analyses based on a concatenated supermatrix (Table 2).

For the *Wolbachia II* data set, we downloaded (if available) proteomes for all available *Wolbachia* draft and fully assembled genomes (47 in total, see supplementary material for details). Proteins for *Wolbachia* strains which were lacking this information on *NCBI GenBank* were derived from translations using *GeneMark* version 2.5 [3]. We predicted groups of orthologous genes between these proteomes using *Orthofinder* version 2.1.2 [17] running under default parameters. Single copy genes present in all analysed strains (83 in total) were aligned using *MAFFT* version 7.271 with the L-INS-i algorithm [37], and tested for evidence of recombination using the pairwise homoplasy index (*PHI*) [8] with window sizes of 10, 20, 30, and 50. Recombining loci were subsequently removed from the data set and the remaining loci concatenated using *FasConCat* version 1.0 [39]. The resulting supermatrix (68 loci, 20,787 amino acid positions) was subject to partitioned *Maximum Likelihood* analysis following best model and partition scheme selection in *IQ-TREE* version 1.6.2 [10, 36, 49];

For the whole-proteome sequences of the data set *Wolbachia I*, the *RF* distance to the reference tree was 6 for each of the 100 program runs. By contrast, for the 100 runs on the selected protein sequences of the same set of taxa, the average *RF* distance was 7.68, the standard deviation was 0.736. For *Wolbachia II*, the average *RF* distance was 19.62; the standard deviation was 0.89.

Large-scale microbial phylogeny reconstruction

In 2013, J. Eisen's group published a paper on the phylogeny of the microbial genomes that were available at the time [40]. As a basis of their study, they selected 24 single-copy marker genes and a non-redundant subset of taxa. To obtain such a subset, they used a greedy algorithm by M. Steel [64], making sure that marker genes from different taxa in the resulting subset had a distance to each other of at least 2 substitutions per 100 positions. This way, they obtained a non-redundant subset of 841 bacterial and archeal genomes from the more than 3,000 microbial genomes that were publicly available. Multiple sequence alignments of the marker genes were calculated with *hmmalign* [16] and were concatenated to a *supermatrix* which was used

as input for the phylogeny programs *RAxML* [62] and *MrBayes* [57]. In addition, the authors used the Bayesian tree-reconciliation program *BUCKy* [41] to the same set of marker genes. The trees they obtained with these different methods were found to be similar to trees obtained based on *16S RNA* genes.

To evaluate *Prot-SpaM*, we used the 841 microbial genomes from Lang *et al.* [40]. and downloaded all protein sequences from these taxa that were available through *GenBank*. For 28 out of the 841 taxa, we were unable to obtain protein sequences, so we obtained a slightly reduced subset of 813 taxa, compared to the taxa used by Lang *et al.* First, we applied *Prot-SpaM* to all available protein sequences from these 813 taxa. In addition, we ran *Prot-SpaM* on the protein sequences encoded by the 24 marker genes from Lang *et al.* and, finally, we applied our previous approach *Filtered Spaced Word Matches* [44] to the 841 genome sequences. The trees that we obtained with our different alignment-free approaches are shown in Figure 6, together with the *Maximum Likelihood* tree from [40] which we considered as a reliable reference. Clades from this reference tree are color-coded in Figure 6. As can be see from the color coding, the tree obtained with *Prot-SpaM* from the available protein sequences contains essentially the same clades as the reference tree. There are some differences within the clades, though, that should be further investigated (J. Eisen, personal communication). The *RF* distance between the tree obtained with *Prot-SpaM* and the reference tree was 1,020.

Plants

Next, we used a set of plant taxa that has been previously studied by Hatje and Kollmar [26] and that we had already used in previous studies to evaluate alignment-free approaches to genome sequence comparison [42–44]. The data set that we used in these previous papers consisted of 14 brassicales species. In *GenBank*, however, the proteomes could be downloaded only for 11 of the 14 species, so we had to limit our test runs to these 11 species. To obtain a reference tree, we used a tree that has been obtained with multiple sequence alignment and maximum likelihood as published by Hatje and Kollmar [26], Figure 3 B. From this tree, we removed the three species for which we could not obtain the proteome sequences in *GenBank*. Figure 7 shows the reference tree of the 14 original species, together with trees of the 11 species with available proteomes, calculated with the alignment-free methods that we evaluated in this paper. For the 100 program runs with *Prot-SpaM*, the average *RF* distance between the resulting trees and the

	# taxa	total size [MB]	Source
E. coli / Shigella	29	56.41	Zhou *et al.* [74]
Wolbachia I, 252 proteins	19	1.15	Gerth *et al.* [23]
Wolbachia I, whole proteomes	19	7.96	Gerth *et al.* [23]
Wolbachia II	47	14.78	See supplementary material
Plants	11	245.05	Hatje and Kollmar [26]
Prokaryotes	813	784.86	Lang *et al.* [40]
Metazoa	36	585.0	Borowiec *et al.* [6]

Table 1: Data sets used in this study to evaluate alignment-free methods, with number of taxa, total size and source of the reference tree.

reference tree from [26] was 0.82; the standard deviation was 0.9.

Metazoa

Finally, we used a set of 36 proteomes from 34 *metazoan* and two choanoflagellate taxa. These taxa have been previously used by Borowiec *et al.* [6] to study the position of the Ctenophora within the phylogenetic tree of the metazoan kingdom. The same set of taxa has also been used in a study by Zhou *et al* [73] to evaluate *Maximum-Likelihood* programs for phylogeny reconstruction. As a reference tree, we used the tree published in [6]. The average *RF* distance of the *Prot-SpaM* trees to this reference tree was 27.1, with a standard deviation of 1.51.

Parameter values and number of selected spaced-word matches

Prot-SpaM has four major parameters which can be adjusted by the user: the *weight* w (=number of *match positions*) of the binary patterns and spaced words, their *length* ℓ, the number m of different binary patterns used by the program and the cut-off value T to separate homologous from background spaced-word matches. To see how these parameters influence the results of our software, and to find suitable default values, we ran *Prot-SpaM* with

	Prot-SpaM	*FSWM*	*CVTree*	*FFP*, k = 4	*kmacs*, k = 10	*ACS*
			RF distances			
E. coli / Shigella	4.00	6	24	40	42	38
Wolbachia I, 252 proteins	7.68	8	6	4	8	4
Wolbachia I, whole proteomes	6.00	6	8	16	8	12
Wolbachia II	19.62	20	44	54	26	16
Plants	0.82	0	6	8	2	6
Prokaryotes	1,020	1,348	886	1,452	880	960
Metazoa	27.1	-	40	62	30	36
			Relative *RF* distances			
E. coli / Shigella	0.08	0.12	0.46	0.77	0.81	0.73
Wolbachia I, 252 proteins	0.24	0.25	0.19	0.13	0.25	0.13
Wolbachia I, whole proteomes	0.19	0.19	0.25	0.50	0.25	0.38
Wolbachia II	0.23	0.23	0.50	0.61	0.30	0.18
Plants	0.05	0.00	0.37	0.50	0.12	0.37
Prokaryotes	0.63	0.83	0.55	0.90	0.54	0.59
Metazoa	0.41	-	0.61	0.94	0.45	0.55

Table 2: *Robinson-Foulds (RF)* distances and *relative RF* distances between trees generated with alignment-free methods and the respective reference trees for various sets of taxa, see the main text for details. Since *Prot-SpaM* uses a probabilistic algorithm, different program runs may produce slightly different results. Therefore, we performed 100 program runs on each data set and report the *average RF* distances, except for the large prokaryote data set where we did only one single program run. All programs were run on *protein* sequences or *whole proteomes*, respectively, except for *Filtered Spaced Word Matches (FSWM)*, which was run on *whole-genome sequences* of the same species (or on the gene sequences coding for the 252 selected proteins from *Wolbachia I*). We were unable to run *FSWM* on the whole genomes of the 31 *metazoan* species, since this data set was too large. Since the original implementation of *ACS* is not publicly available, we ran our own implementation, *kmacs*, with $k = 0$ instead.

	Prot-SpaM	*FSWM*	*CVTree*	*FFP*, k = 4	*kmacs*, k = 10	*ACS*
E. coli / Shigella	55	110	125	10	2,518	193
Wolbachia II	19	68	46	9	5,302	135
Wolbachia I, 252 proteins	3	5	2	1	36	3
Wolbachia I, whole proteomes	11	22	21	2	178	26
Plants	464	1,107,720	365	17	17,693	850
Prokaryotes	5,502	244,139	5,492	1,929	915,635	123,520
Metazoa	1,719	-	1,973	43	151,612	9,512

Table 3: Program run time in seconds for different alignment-free approaches on our benchmark data sets. *Prot-SpaM* and *FSWM* were run on 40 threads. The other tools do not support multi-threading, therefore they were run single threaded.

varying values of these four parameters. Here, we modified one parameter at a time, using the respective default values of the remaining three parameters. The results are summarized in Tables 4 and 5. As can be seen from these tables, there are no values for w, ℓ an T that work best for all data sets, but our default values seem to be a reasonable compromise. Using sets of $m = 5$ binary patterns does not improve the quality of the produced trees in terms of their *RF* distances to the reference trees, compared to program runs with single patterns. Table 3 shows, however, that the distance values estimated by *Prot-SpaM* become statistically more stable if multiple patterns are used.

The number of spaced-word matches in a pairwise sequence comparison depends on how similar the two sequences are to each other, see [48] for details. Consequently, the number of spaced-word matches that are selected by our program to estimate phylogenetic distances also depends on the degree of similarity between the compared sequences. We found two extreme cases with our test data, one in the *E. coli/Shigella* data set where most taxa are closely related to each other, and another one in the *Metazoan* data set that contains taxa with very large evolutionary distances. In the pairwise comparison of *E. coli O157:H7 strain EDL933* with *E. coli O157:H7 Sakai*

(EHEC), *Prot-SpaM* selected more than 6,000,000 spaced-word matches. These two proteomes have less than 1,600,000 amino acids each, so in this case > 3.75 spaced-word matches per sequence position were selected. By contrast, less than 13.000 spaced-word matches were selected in the comparison of *Brugia malayi* and *Homo sapiens*. The latter proteome has a length of more than 75.000.000 amino acids, so here less than 0.00017 spaced-word matches per sequence position were selected.

4 Discussion

A number of so-called 'alignment-free' approaches have been proposed in recent years to rapidly calculate phylogenetic distances between genomic sequences. Earlier approaches are based on k-mer frequencies or on the length of common substrings; these approaches have been applied not only to DNA, but also to protein sequences. A draw-back of these methods is that they can only calculate rough measures of sequence similarity or dissimilarity, they do not estimate phylogenetic distances in a rigorous way. More recently, word-based methods have been developed that can accurately estimate phylogenetic distances between genomic sequences based on stochastic models of DNA evolution. One of these approaches is *Filtered-Spaced Word Matches (FSWM)*.

In this study, we introduced *Prot-SpaM*, a new implementation of *FSWM* to compare complete or incomplete *proteome sequences* to each other. To our knowledge, *Prot-SpaM* is the first tool that can accurately estimate phylogenetic distances between protein sequences without the need to calculate full sequence alignments. Our benchmark results show that distance estimates obtained with our approach are accurate for a large range of phylogenetic distances. Distances calculated with *CVTree, ACS, FFP* and *kmacs*, by comparison, are monotonously increasing with the number of substitutions between the compared sequences. The obtained distance values are far from proportional to the real distances, though, and they flatten out somewhere between 0.5 and 1.5 substitutions per position, see Figure 2. By contrast, *Prot-SpaM* estimates distances with high accuracy for up to around 2.0 substitutions per position. For higher distance values, the calculated distances become less stable, as can be seen from the error bars in Figure 2. Moreover, for large distances, our program tends to slightly overestimate distances.

In our program evaluation, we used all competing software tools with their respective default parameters, if such default values were recommended by their developers. It should be mentioned, however, that some of the eval-

E. coli/Shigella								
Weight w		**6**	8	10				
Runtime [s]		**55.4**	47.4	46.9				
RF distance		**4**	6.02	6.76				
Length ℓ	36	**46**	56	66				
Runtime [s]	47.5	**55.4**	60.3	66.9				
RF distance	4	**4**	4.02	4.84				
# patterns m	1	3	**5**	7				
Runtime [s]	13.5	34.4	**55.4**	75.94				
RF distance	4.12	4.02	**4**	4				
Threshold T	-50	-25	**0**	25	50	75	100	125
Runtime [s]	55.2	55.4	**55.4**	55.1	55.3	55.3	55.1	55.2
RF-Distance	11.92	12	**4**	12	12	12	12	12
Wolbachia II								
Weight w	4	**6**	8	10				
Runtime [s]	112.4	**19.4**	18	17.8				
RF distance	20.38	**19.68**	22	22				
Length ℓ	36	**46**	56	66				
Runtime [s]	17	**19.4**	21.5	23.9				
RF distance	19.78	**19.68**	19.7	19.78				
# patterns m	1	3	**5**	7				
Runtime [s]	5.4	12.5	**19.4**	26.5				
RF distance	19.06	19.12	**19.68**	19.82				
Threshold T	-50	-25	**0**	25	50	75	100	125
Runtime [s]	19.6	19.6	**19.4**	19.4	19.4	19.4	19.4	19.4
RF-Distance	22.18	18	**19.68**	19.86	20.16	20.7	21.04	22

Table 4: Program runtime and *Robinson-Foulds* distances to reference trees for different parameter values with *Prot-SpaM* for the *E. coli/Shigella* and *Wolbachia* proteomes. We ran our program with different values for the *weight* w and length ℓ of the spaced-words, for different numbers of patterns and for different values of the threshold T. Here, we modified the value of one of these parameters at a time and used the default values for the remaining three parameters. Default values of the modified parameters and the resulting runtimes and *RF* distances are shown in bold font. Since *Prot-SpaM* uses a probabilistic algorithm to generate pattern sets, we performed 100 program runs for each set of parameters; the table reports the average *RF* distances of these 100 runs.

Plants

Weight w	4	**6**	8	10				
Runtime [s]	57,578	**464**	320	325				
RF distance	2	**0**	4	6				
Length ℓ	36	**46**	56	66				
Runtime [s]	383	**464**	441	494				
RF distance	2	**0**	0	0				
# patterns m	1	3	**5**	7				
Runtime [s]	91	255	**464**	572				
RF distance	0	0	**0**	2				
Threshold T	-50	-25	**0**	25	50	75	100	125
Runtime [s]	383	402	**464**	409	391	459	439	430
RF-Distance	2	2	**0**	0	0	0	2	4

Metazoa

Weight w		**6**	8	10				
Runtime [s]		**1719**	1584	1518				
RF distance		**30**	26	30				
Length ℓ	36	**46**	56	66				
Runtime [s]	1351	**1719**	1584	2089				
RF distance	26	**30**	24	26				
# patterns	1	3	**5**	7				
Runtime [s]	427	890	**1719**	2078				
RF distance	24	28	**30**	26				
Threshold T	-50	-25	**0**	25	50	75	100	125
Runtime [s]	2539	2337	**1719**	2269	2150	1906	1783	1797
RF-Distance	30	24	**30**	26	28	28	30	34

Table 5: Program runtime and *Robinson-Foulds* distances to reference trees for different parameter values with *Prot-SpaM* for the *plant* and *metazoan* proteomes; parameter values as in Table 4. Because of the size of these data sets, we performed only one program run per parameter set.

uated programs might produce better results with different parameter settings. The program *FFP*, for example, uses a default k-mer length of $k = 4$, so we used this value in our study. It has been reported, however, that *FFP* may perform better on protein sequences if larger values of k are used [35]. A comprehensive investigation of the effects of different parameters on the software programs evaluated in this study would be beyond the scope of the present paper. Interested readers are encouraged to run these programs with different parameter values so see if their results on our benchmark data can be improved.

Prot-SpaM produced high-quality trees and was superior to other alignment-free methods for the *E.coli/Shigella* and the plant data sets, as shown in Table 2. On the *Wolbachia* data sets, it still performed reasonably well and was again superior to competing approaches on the whole-proteome sequences, but it was outperformed by word-frequency methods on the 252 selected orthologous proteins. A possible explanation of this result is discussed below. On the large prokaryote data set and for the metazoan set, by contrast, none of the compared programs could reproduce the reference trees that we used in our evaluation. These are difficult data sets since they span very large evolutionary distances; also it should be mentioned that there are no absolutely reliable reference trees available for these data sets. For the metazoan data set, for example, the positions of the ctenophores is still a matter of debate [15,21,51]. On the metazoans, *Prot-SpaM* performed better than other alignment-free approaches, while on the large prokaryote data set, *CVTree, ACS* and *kmacs* were superior.

An interesting result is the performance of *Prot-SpaM*, compared to our previous approach *FSWM* that takes genomic sequences as input. For most groups of taxa in our study, the results of *Prot-SpaM* and *FSWM* were of similar quality, in the sense that the *RF* distances to the reference trees were comparable for both approaches. However, for the set of 813 prokaryote taxa, our new spaced-words approach performed better on whole-proteomes than our previous approach on whole genomes, as is shown in Figure 6 and Table 2. This discrepancy is most likely due to the large phylogenetic distances in this data set; for such distantly related sequences, homologies are generally better detectable at the protein level than at the DNA level.

'Alignment-free' methods to phylogeny reconstruction can be directly applied to whole-genome or whole-proteome sequences, without the need to select orthologous genes or proteins in a first step. This is generally seen as an advantage over more traditional, alignment-based approaches, since the task of finding orthologs is time-consuming and often involves manual intervention. On our data set *Wolbachia I*, we actually obtained better

RF distances with *Prot-SpaM* and *FSWM* when we applied these programs to the whole-protein or whole-genome sequences, respectively, than when we applied them to the 252 selected orthologous proteins or to the genes coding for those proteins, see Table 2. These results are in contrast to the more traditional alignment-free methods *FFP*, *CVtree* and *ACS* that are based on word-frequencies or on the length of common substrings. The latter programs performed better on the selected orthologous proteins of the *Wolbachia I* data set than on the corresponding whole-proteome sequences.

A possible explanation of this phenomenon is that *Prot-SpaM* and *FSWM* can reliably distinguish between homologous and background spaced-word matches and use only homologous matches for phylogenetic inference. With the *one-to-one* spaced-word matching, they can also reduce the number of *paralogous* spaced-word matches. Therefore, they can be applied to whole proteomes or whole genomes without being too much confused by paralogs or by non-related parts of the sequences. Here, the benefits of using larger input sequence sets seem to outweigh the disadvantage of including possible non-related sequences, paralogs or sequences with recombinations. Previously introduced word-frequency or substring-length methods, by contrast, do not distinguish between homologous and non-homologous parts of the sequences. Therefore, these approaches tend to be confused by input sequences that contain paralogs or are only locally related to each other.

Table 3 shows that the run time of Prot-SpaM is superior to *CVtree* and *ACS* on protein sequences. By far the fastest alignment-free method on whole proteomes was *FFP*, the slowest one was *kmacs*. On the plant proteomes, *Prot-Spam* was three orders of magnitude faster than *FSWM* on the genome sequences of the same species. This is not surprising, given the fact that in eukaryotes only a small part of the genome is protein-coding sequence. The total size of the 11 plant genomes was 3.8 *GB*, compared to 245 *MB* for the corresponding proteome sequences (note that, for the genome sequences, both strands are considered and the number of background spaced-word matches scales quadratically with the sequence length).

Prot-SpaM has four major parameters that can be adjusted by the user: the weight w and the length ℓ of the patterns and spaced-words, respectively, the cut-off value T to distinguish homologous from random spaced-word matches and the number m of different patterns used to generate spaced-word matches. We provide default values for these parameters, but Tables 4 and 5 show that reasonable results can be obtained with a rather broad range of parameter values. These tables also show that the quality of the produced trees, as measured by the *RF* distances to the reference trees, could not be improved by using $m = 5$ patterns, compared to the single-pattern

option, *i.e.* $m = 1$. The statistical stability of our distance estimates, however, is increased if multiple patterns are used; therefore, we are using $m = 5$ patterns by default. But since runtime and memory usage of our program increase with the number m of pattern, it may be advisable to use the single-pattern option if very large data sets are to be analyzed.

It should be mentioned that traditional approaches to phylogeny reconstruction that are based on multiple sequence alignment are still more accurate than alignment-free approaches that have been proposed in recent years. The main advantage of these novel approaches is their high speed, with makes it possible to apply them to the large sequence data sets that are now available; a program run of *Prot-SpaM* on whole-proteome sequences of the set *Wolbachia II* that consists of 47 taxa, took only 19 seconds. Another advantage of our approach is that it can reliably distinguish between local homologies and random background similarities. It can, thus, be applied to complete or incomplete proteomes, and it is not necessary to select orthologous genes or proteins in a first step. Therefore, we think that *Prot-SpaM* should be a useful addition to existing approaches to phylogeny reconstruction.

Availability of source code and requirements

- Project name: Prot-SpaM
- Project home page: https://github.com/jschellh/ProtSpaM
- Operating system(s): linux
- Programming language: C++
- Other requirements: none
- License: GNU GPL
- Any restrictions to use by non-academics: none

Availability of supporting data

The sequence data sets and trees used for the program evaluation can be downloaded here: http://projects.gobics.de/data/protspam/paperData.tgz

Abbreviations

Prot-Spam: Proteome-based Spaced-Word Matches; FFP: Feature Frequency Profile; FSWM: Filtered Spaced-Word Matches; spamogram: Spaced-Word-Match Histogram; PAM: Point Accepted Mutation; BLOSUM: Blocks Substitution Matrix; rasbhari: Rapid Approach for Seed optimization Based on a Hill-climbing Algorithm that is Repeated Iteratively; ACS: Average Common Substring Aproach; kmacs: k-Mismatch Average Common Substring Approach; CVTree: Composition Vector Tree; RF: Robinson-Foulds; iTOL: Interactive Tree Of Life; PHYLIP: PHYLogeny Inference Package; MAFFT: multiple alignment using fast Fourier transform; PHI: Pairwise Homoplasy Index; RAxML: Randomized Axelerated Maximum Likelihood; GB: Gigabase; MB: Megabase;

Competing interests

The authors declare that they have no competing interests

Consent for publication

Not applicable

Funding

The project was partially funded by the VW Foundation, project VWZN3157. We acknowledge support by the *Open Access Publication Funds of the Göttingen University.*

Author contributions

CAL designed the study, co-supervised the project and drafted parts of the manuscript, JS implemented the software and did most of the program evaluation, SD contributed to the program evaluation, MG and CB analysed and discussed the results on Wolbachia and wrote parts of the manuscript, BM co-supervised the project and wrote most of the manuscript.

Acknowledgements

We thank Jonathan Eisen for his comments on the phylogeny of the 841 microbial taxa that were studied in Lang *et al.* Five reviewers made very helpful comments on an earlier version of this manuscript. We acknowledge support by the *Open Access Publication Funds* of the *Göttingen University.*

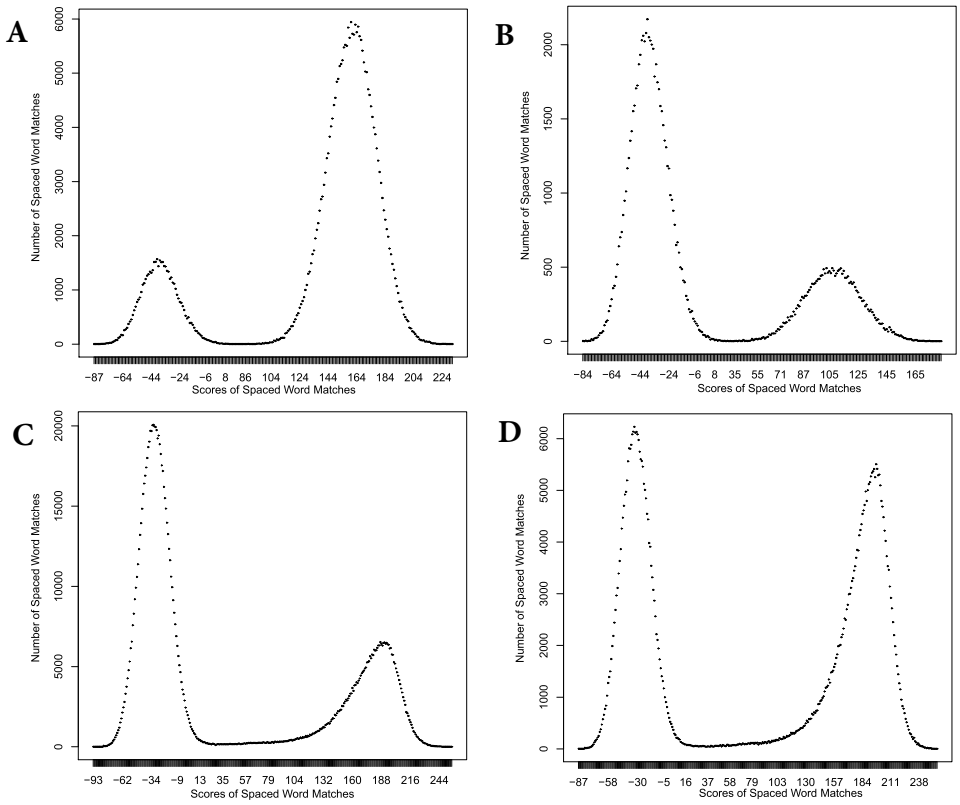

Figure 1: *Spaced-word histograms ('spamograms')* for different data sets. (A) and (B) are based on simulated indel-free protein sequences with a total length of of 1.6×10^6 amino-acid residues each, and with 0.3 (A) and 0.75 (B) substitutions per position, respectively. (C) and (D) are from a whole-proteome comparisons of plants, (C) comparing *Eucalyptus grandis* with *Capsella rubella* and (D) comparing *Gossypium raimondii* with *Carica papaya*.

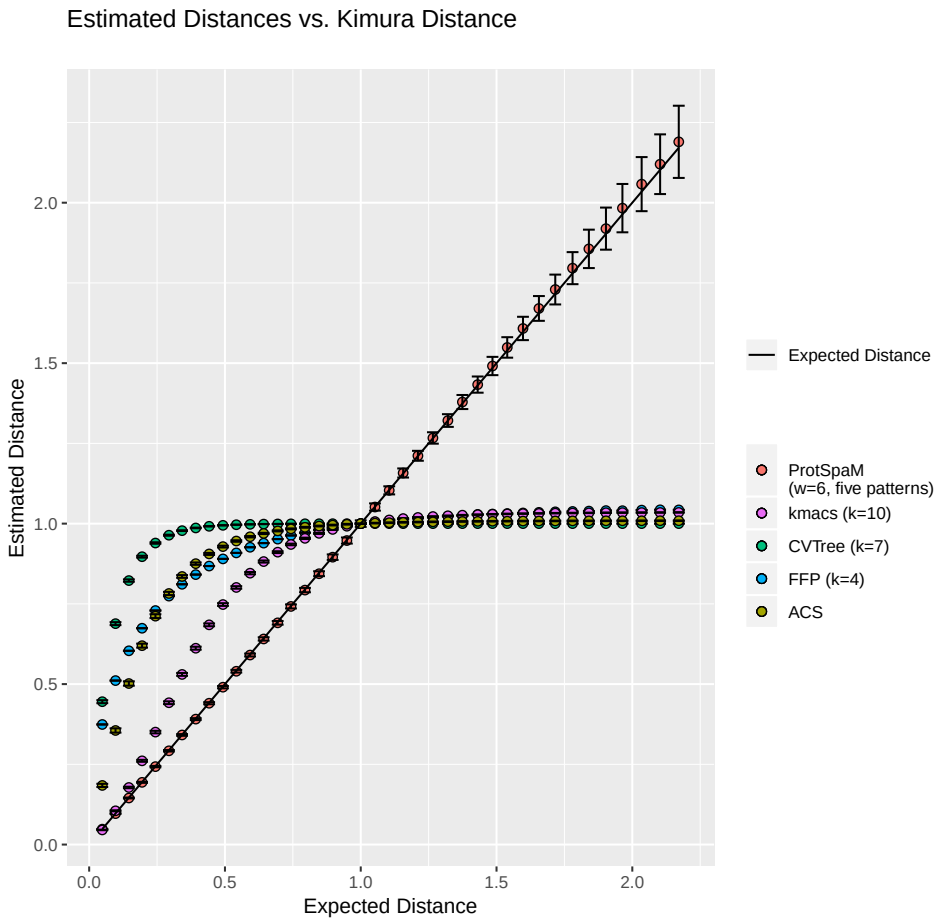

Figure 2: Distances calculated by *Prot-SpaM* and four other alignment-free methods calculated for pairs of simulated protein sequences, plotted against their distances calculated with the *Kimura* model. Error bars denote standard deviations. Note that *Prot-SpaM* estimates phylogenetic distances in terms of substitutions that have happened since two sequences evolved from their lasts common ancestor. The programs *kmacs, CVTree, FFP* and *ACS*, by contrast, do not estimate distances in a rigorous way, but rather use ad-hoc measures of sequence dissimilarity that are not linear functions of the real distances. Also, the absolute values of these distance measures are rather arbitrary for these four other programs. We therefore normalized the distances calculated by *kmacs, CVTree, FFP* and *ACS* such that they have a value of one for sequence pairs with a *Kimura* distance of one.

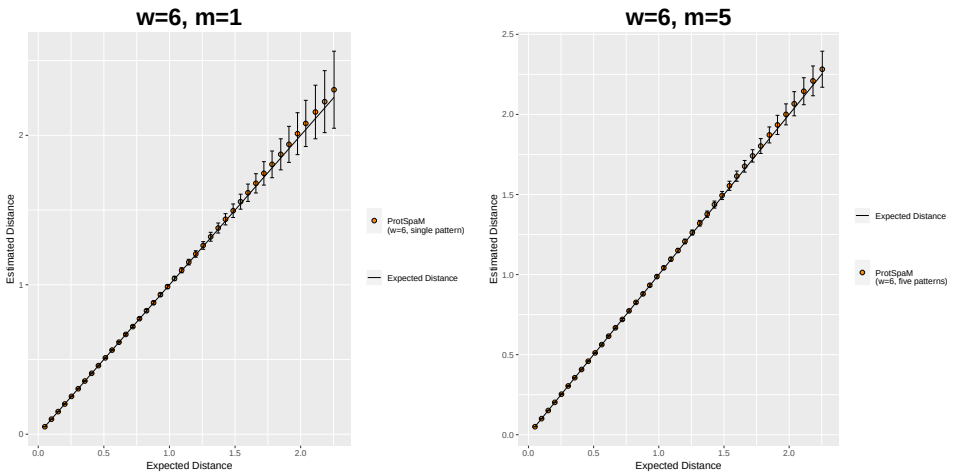

Figure 3: Distances calculated by *Prot-SpaM* for pairs of simulated protein sequences with a single binary pattern ($m = 1$, left) and with the default multiple-pattern option ($m = 5$, right). We performed 1000 program runs for each value of m. The plot shows the *average* of the calculated distances; *standard deviations* are shown as error bars.

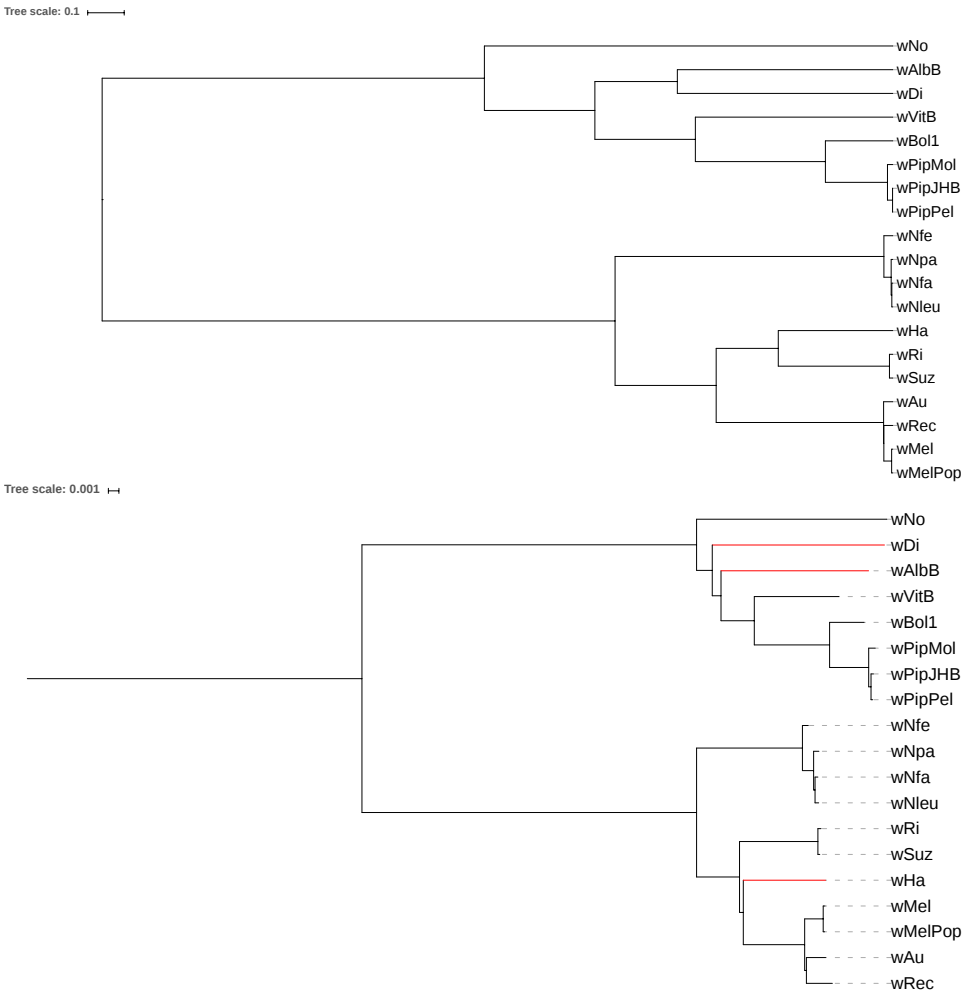

Figure 4: Reference tree for our data set *Wolbachia I* (above) and tree calculated with *Prot-SpaM* using whole-proteome sequences of the same taxa (below), see main text for details. Topological differences between the two trees are shown in red in the *Prot-SpaM* tree.

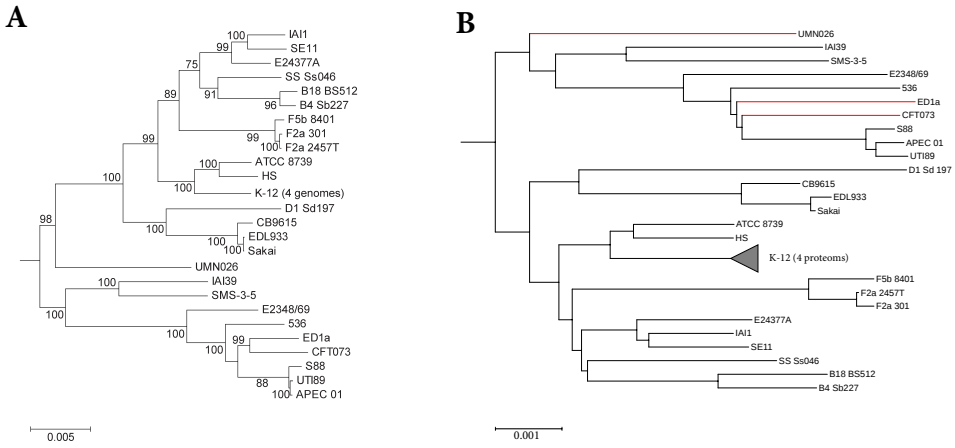

Figure 5: Reference tree (A) from [74] and tree calculated with *Prot-SpaM* with default parameters (B) for a set of 29 *Escherichia coli* and *Shigella* strains. Differences in the topologies between the two trees are marked in red.

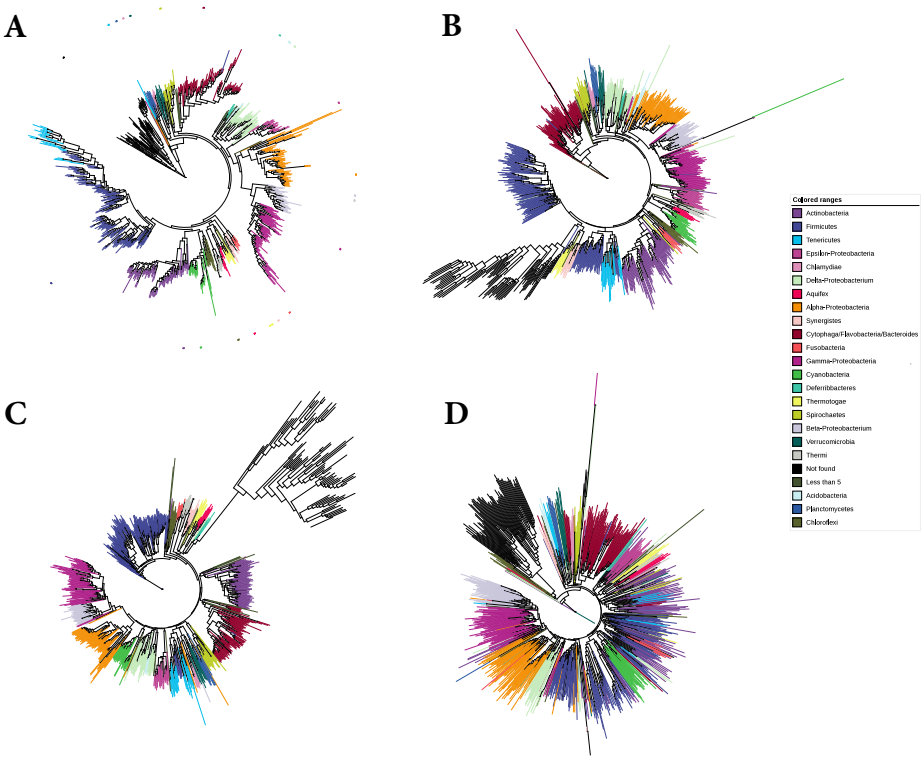

Figure 6: Phylogenetic trees for a large set of microbial taxa studied by Lang *et al.* [40]. (A) Maximum-Likelihood tree constructed by Lang *et al.* based on a super alignment of 24 selected genes, (B) tree constructed with our approach, as described in this paper, for 813 taxa for which the proteomes are available in *GenBank*, (C) tree constructed with our approach based on the proteins corresponding to the 24 genes selected by Lang *et al.* and (D) tree reconstructed using our program *FSWM* [44] on the 841 whole-genome sequences.

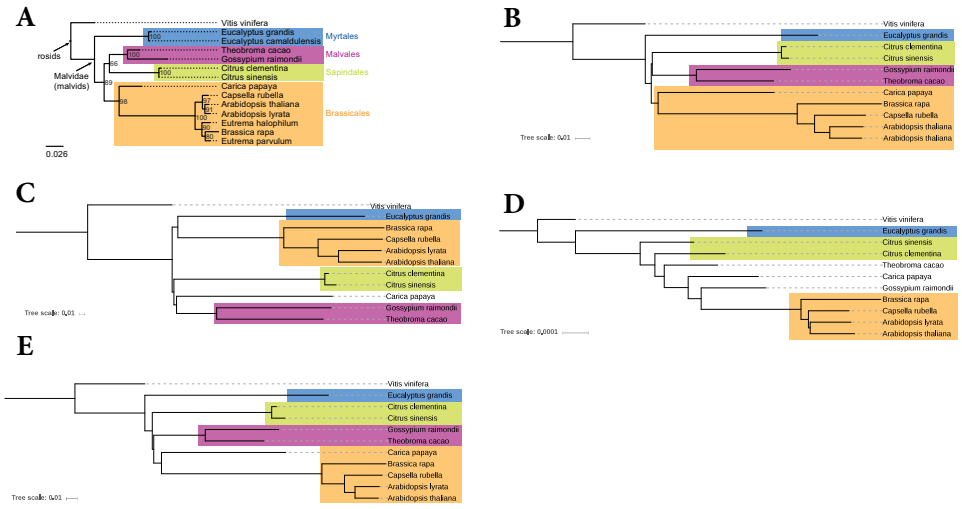

Figure 7: Phylogenetic trees of plant taxa. (A) reference tree from [26], and trees constructed with (B) the approach described in this paper, (C) *ACS* [68], (D) *FFP* [59], and (E) *kmacs* [43]. The original data set contained 14 taxa, but only for 11 taxa, the proteomes could be downloaded through *GenBank*. For completeness, we show the reference for all 14 taxa.

References

[1] Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, James Ostell, Kim D Pruitt, and Eric W Sayers. Genbank. *Nucleic Acids Research*, 46(D1):D41–D47, 2018.

[2] Guillaume Bernard, Cheong Xin Chan, Yao-ban Chan, Xin-Yi Chua, Yingnan Cong, James M. Hogan, Stefan R. Maetschke, and Mark A. Ragan. Alignment-free inference of hierarchical and reticulate phylogenomic relationships. *Briefings in Bioinformatics*, in press:bbx067, 2017.

[3] John Besemer and Mark Borodovsky. GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Research*, 33:W451–W454, 2005.

[4] Olaf R.P. Bininda-Emonds. The evolution of supertrees. *Trends in Ecology and Evolution*, 19:315 – 322, 2004.

[5] Seth R. Bordenstein, Charalampos Paraskevopoulos, Julie C. Dunning Hotopp, Panagiotis Sapountzis, Nathan Lo, Claudio Bandi, Herv Tettelin, John H. Werren, and Kostas Bourtzis. Parasitism and mutualism in Wolbachia: What the phylogenomic trees can and cannot say. *Molecular Biology and Evolution*, 26:231–241, 2009.

[6] Marek L. Borowiec, Ernest K. Lee, Joanna C. Chiu, and David C. Plachetzki. Extracting phylogenetic signal and accounting for bias in whole-genome data sets supports the Ctenophora as sister to remaining Metazoa. *BMC Genomics*, 16:987, 2015.

[7] Amanda M. V. Brown, Sulochana K. Wasala, Dana K. Howe, Amy B. Peetz, Inga A. Zasada, and Dee R. Denver. Genomic evidence for plant-parasitic nematodes as the earliest Wolbachia hosts. *Scientific Reports*, 6:34955, 2016.

[8] Trevor C. Bruen, Hervé Philippe, and David Bryant. A simple and robust statistical test for detecting the presence of recombination. *Genetics*, 172:2665–2681, 2006.

[9] David Bryant and Mike Steel. Computing the distribution of a tree metric. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 6:420–426, 2009.

[10] Olga Chernomor, Arndt von Haeseler, and Bui Quang Minh. Terrace aware data structure for phylogenomic inference from supermatrices. *Systematic Biology*, 65:997–1008, 2016.

[11] Benny Chor, David Horn, Yaron Levy, Nick Goldman, and Tim Massingham. Genomic DNA k-mer spectra: models and modalities. *Genome Biology*, 10:R108, 2009.

[12] Matteo Comin and Davide Verzotto. Alignment-free phylogeny of whole genomes using underlying subwords. *Algorithms for Molecular Biology*, 7:34, 2012.

[13] Margaret O. Dayhoff, Robert M. Schwartz, and Bruce C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 6:345–362, 1978.

[14] Thomas Dencker, Chris-André Leimeister, Michael Gerth, Christoph Bleidorn, Sagi Snir, and Burkhard Morgenstern. Multi-SpaM: a Maximum-Likelihood approach to phylogeny reconstruction using multiple spaced-word matches and quartet trees. In Mathieu Blanchette and Aïda Ouangraoua, editors, *Comparative Genomics*, pages 227–241. Springer International Publishing, 2018.

[15] Casey W. Dunn, Gonzalo Giribet, Gregory D. Edgecombe, and Andreas Hejnol. Animal phylogeny and its evolutionary implications. *Annual Review of Ecology, Evolution, and Systematics*, 45:371–395, 2014.

[16] Sean R. Eddy. A new generation of homology search tools based on probabilistic inference. In *Genome Informatics 2009 - Proceedings of the 20th International Conference*, pages 205–211. Imperial College Press, 2009.

[17] David M. Emms and Steven Kelly. OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biology*, 16:157, 2015.

[18] Huan Fan, Anthony R. Ives, Yann Surget-Groba, and Charles H. Cannon. An assembly and alignment-free method of phylogeny reconstruction from next-generation sequencing data. *BMC Genomics*, 16:522, 2015.

[19] Joseph Felsenstein. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, 5:164–166, 1989.

[20] Umberto Ferraro-Petrillo, Gianluca Roscigno, Giuseppe Cattaneo, and Raffaele Giancarlo. Informational and linguistic analysis of large genomic sequence collections via efficient hadoop cluster algorithms. *Bioinformatics*, 34:18261833, 2018.

[21] Roberto Feuda, Martin Dohrmann, Walker Pett, Herv Philippe, Omar Rota-Stabelli, Nicolas Lartillot, Gert Wrheide, and Davide Pisani. Improved modeling of compositional heterogeneity supports sponges as sister to all other animals. *Current Biology*, 27:3864 – 3870.e4, 2017.

[22] Michael Gerth and Christoph Bleidorn. Comparative genomics provides a timeframe for *Wolbachia* evolution and exposes a recent biotin synthesis operon transfer. *Nature Microbiology*, 2:16241, 2016.

[23] Michael Gerth, Marie-Theres Gansauge, Anne Weigert, and Christoph Bleidorn. Phylogenomic analyses uncover origin and spread of the Wolbachia pandemic. *Nature Communications*, 5:5117, 2014.

[24] Eliza Glowska, Anna Dragun-Damian, Miroslawa Dabert, and Michael Gerth. New Wolbachia supergroups detected in quill mites (Acari: Syringophilidae). *Infection, Genetics and Evolution*, 30:140–146, 2015.

[25] Lars Hahn, Chris-André Leimeister, Rachid Ounit, Stefano Lonardi, and Burkhard Morgenstern. *rasbhari*: optimizing spaced seeds for database searching, read mapping and alignment-free sequence comparison. *PLOS Computational Biology*, 12(10):e1005107, 2016.

[26] Klas Hatje and Martin Kollmar. A phylogenetic analysis of the brassicales clade based on an alignment-free sequence comparison method. *Frontiers in Plant Science*, 3:192, 2012.

[27] Bernhard Haubold. Alignment-free phylogenetics and population genetics. *Briefings in Bioinformatics*, 15:407–418, 2014.

[28] Bernhard Haubold, Fabian Klötzl, and Peter Pfaffelhuber. andi: Fast and accurate estimation of evolutionary distances between closely related genomes. *Bioinformatics*, 31:1169–1175, 2015.

[29] Bernhard Haubold, Peter Pfaffelhuber, Mirjana Domazet-Loso, and Thomas Wiehe. Estimating mutation distances from unaligned genomes. *Journal of Computational Biology*, 16:1487–1500, 2009.

[30] Steven Henikoff and Jora G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.

[31] Michael Höhl, Isidore Rigoutsos, and Mark A. Ragan. Pattern-based phylogenetic distance estimation and tree reconstruction. *Evolutionary Bioinformatics Online*, 2:359–375, 2006.

[32] Sebastian Horwege, Sebastian Lindner, Marcus Boden, Klaus Hatje, Martin Kollmar, Chris-André Leimeister, and Burkhard Morgenstern. *Spaced words* and *kmacs*: fast alignment-free sequence comparison based on inexact word matches. *Nucleic Acids Research*, 42:W7–W11, 2014.

[33] Lucian Ilie, Silvana Ilie, and Anahita M. Bigvand. SpEED: fast computation of sensitive spaced seeds. *Bioinformatics*, 27:2433–2434, 2011.

[34] David T. Jones, William R. Taylor, and Janet M. Thornton. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282, 1992.

[35] Se-Ran Jun, Gregory E. Sims, Guohong A. Wu, and Sung-Hou Kim. Whole-proteome phylogeny of prokaryotes by feature frequency profiles: An alignment-free method with optimal feature resolution. *Proceedings of the National Academy of Sciences*, 107:133–138, 2010.

[36] Subha Kalyaanamoorthy, Minh Q. Bui, Thomas K F Wong, Arndt von Haeseler, and Lars S. Jermiin. ModelFinder: fast model selection for accurate phylogenetic estimates. *Nature Methods*, 14:587–589, 2017.

[37] Kazutaka Katoh and Daron M. Standley. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, 30:772–780, 2013.

[38] Motoo Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983.

[39] Patrick Kück and Gary C. Longo. FASconCAT-G: extensive functions for multiple sequence alignment preparations concerning phylogenetic studies. *Frontiers in Zoology*, 11:81, 2014.

[40] Jenna Morgan Lang, Aaron E. Darling, and Jonathan A. Eisen. Phylogeny of bacterial and archaeal genomes using conserved genes: Supertrees and supermatrices. *PLOS ONE*, 8:e62510, 2013.

[41] Bret R. Larget, Satish K. Kotha, Colin N. Dewey, and Cécile Ané. BUCKy: Gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics*, 26:2910–2911, 2010.

[42] Chris-André Leimeister, Marcus Boden, Sebastian Horwege, Sebastian Lindner, and Burkhard Morgenstern. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics*, 30:1991–1999, 2014.

[43] Chris-André Leimeister and Burkhard Morgenstern. *kmacs*: the k-mismatch average common substring approach to alignment-free sequence comparison. *Bioinformatics*, 30:2000–2008, 2014.

[44] Chris-André Leimeister, Salma Sohrabi-Jahromi, and Burkhard Morgenstern. Fast and accurate phylogeny reconstruction using filtered spaced-word matches. *Bioinformatics*, 33:971–979, 2017.

[45] Ivica Letunic and Peer Bork. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic Acids Research*, 44:W242–W245, 2016.

[46] Liang Liu, Zhenxiang Xi, Shaoyuan Wu, Charles C. Davis, and Scott V. Edwards. Estimating phylogenetic trees from genome-scale data. *Annals of the New York Academy of Sciences*, 1360:36–53, 2015.

[47] Burkhard Morgenstern, Svenja Schöbel, and Chris-André Leimeister. Phylogeny reconstruction based on the length distribution of k-mismatch common substrings. *Algorithms for Molecular Biology*, 12:27, 2017.

[48] Burkhard Morgenstern, Bingyao Zhu, Sebastian Horwege, and Chris-André Leimeister. Estimating evolutionary distances between genomic sequences from spaced-word matches. *Algorithms for Molecular Biology*, 10:5, 2015.

[49] Lam-Tung Nguyen, Heiko A. Schmidt, Arndt von Haeseler, and Bui Quang Minh. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32:268–274, 2015.

[50] Laurent Noé. Best hits of 11110110111: model-free selection and parameter-free sensitivity calculation of spaced seeds. *Algorithms for Molecular Biology*, 12:1, 2017.

[51] Hervé Philippe, R Derelle, P Lopez, Kerstin Pick, C Borchiellini, N Boury-Esnault, J Vacelet, E Renard, E. Houliston, E. Quéinnec,

C. Da Silva, P. Wincker, H. Le Guyader, S. Leys, Daniel Jackson, Fabian Schreiber, Dirk Erpenbeck, Burkhard Morgenstern, Gert Wörheide, and M Manuel. Phylogenomics restores traditional views on deep animal relationships. *Current Biology*, 19:706–712, 2009.

[52] Cinzia Pizzi. MissMax: alignment-free sequence comparison with mismatches through filtering and heuristics. *Algorithms for Molecular Biology*, 11:6, 2016.

[53] Ji Qi, Hong Luo, and Bailin Hao. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. *Nucleic Acids Research*, 32(suppl 2):W45–W47, 2004.

[54] Gesine Reinert, David Chew, Fengzhu Sun, and Michael S. Waterman. Alignment-free sequence comparison (I): Statistics and power. *Journal of Computational Biology*, 16:1615–1634, 2009.

[55] Jie Ren, Xin Bai, Yang Young Lu, Kujin Tang, Ying Wang, Gesine Reinert, and Fengzhu Sun. Alignment-free sequence analysis and applications. *Annual Review of Biomedical Data Science*, 1:93–114, 2018.

[56] David F Robinson and Les Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[57] Fredrik Ronquist and John P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19:1572–1574, 2003.

[58] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[59] Gregory E. Sims, Se-Ran Jun, Guohong A. Wu, and Sung-Hou Kim. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106:2677–2682, 2009.

[60] Kai Song, Jie Ren, Zhiyuan Zhai, Xuemei Liu, Minghua Deng, and Fengzhu Sun. Alignment-free sequence comparison based on next-generation sequencing reads. *Journal of Computational Biology*, 20:64–79, 2013.

[61] Stephanie J. Spielman and Claus O. Wilke. Pyvolve: A flexible python module for simulating sequences along phylogenies. *PLOS ONE*, 10(9):e0139047, 2015.

[62] Alexandros Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22:2688–2690, 2006.

[63] Alexandros Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30:1312–1313, 2014.

[64] Mike Steel. Phylogenetic diversity and the greedy algorithm. *Systematic Biology*, 54:527–529, 2005.

[65] Hanno Teeling, Jost Waldmann, Thierry Lombardot, Margarete Bauer, and Frank Oliver Glöckner. Tetra: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in dna sequences. *BMC Bioinformatics*, 5:163, 2004.

[66] Sharma V. Thankachan, Sriram P. Chockalingam, Yongchao Liu, and Ambujam Krishnan Srinivas Aluru. A greedy alignment-free distance estimator for phylogenetic inference. *BMC Bioinformatics*, 18:238, 2017.

[67] Sharma V. Thankachan, Sriram P. Chockalingam, Yongchao Liu, Alberto Apostolico, and Srinivas Aluru. ALFRED: a practical method for alignment-free distance computation. *Journal of Computational Biology*, 23:452–460, 2016.

[68] Igor Ulitsky, David Burstein, Tamir Tuller, and Benny Chor. The average common substring approach to phylogenomic reconstruction. *Journal of Computational Biology*, 13:336–350, 2006.

[69] Susana Vinga, Alexandra M. Carvalho, Alexandre P. Francisco, Luís M. S. Russo, and Jonas S. Almeida. Pattern matching through Chaos Game Representation: bridging numerical and discrete data structures for biological sequence analysis. *Algorithms for Molecular Biology*, 7:10, 2012.

[70] Lin Wan, Gesine Reinert, Fengzhu Sun, and Michael S Waterman. Alignment-free sequence comparison (II): theoretical power of comparison statistics. *Journal of Computational Biology*, 17:1467–1490, 2010.

[71] John H. Werren, Laura Baldo, and Michael E. Clark. Wolbachia: master manipulators of invertebrate biology. *Nature Reviews Microbiology*, 6:741 – 751, 2008.

[72] Huiguang Yi and Li Jin. Co-phylog: an assembly-free phylogenomic approach for closely related organisms. *Nucleic Acids Research*, 41:e75, 2013.

[73] Xiaofan Zhou, Xing-Xing Shen, Chris Todd Hittinger, and Antonis Rokas. Evaluating fast maximum likelihood-based phylogenetic programs using empirical phylogenomic data sets. *Molecular Biology and Evolution*, 35:486–503, 2018.

[74] Zhemin Zhou, Xiaomin Li, Bin Liu, Lothar Beutin, Jianguo Xu, Yan Ren, Lu Feng, Ruiting Lan, Peter R. Reeves, and Lei Wang. Derivation of *Escherichia coli* O157:H7 from Its O55:H7 Precursor. *PLOS ONE*, 5:e8700, 2010.

[75] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M. Karlowski. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*, 18:186, 2017.

5 Discussion

In this work, I introduced the *filtered spaced-word matches approach (FSWM)*, a new alignment-free method for fast and accurate genomic phylogeny reconstruction (see Chapter 2). Similar to other methods based on micro-alignments (see Table 1), *FSWM* estimates substitution rates based on short local gap-free alignments. To this end, spaced-word matches are identified and the fraction of non-matching nucleotides at the *don't-care* positions are determined which are turned into an evolutionary distance. To reduce noise from random matches, I proposed a simple but effective filtering procedure to separate background matches and homologous matches. The results showed that this filter removes most of the unwanted background noise and consequently the estimated distances based on the remaining word matches are very accurate. Moreover, I performed a phylogenetic analysis of multiple real-word data sets and showed that the resulting phylogenies were in most cases superior to the phylogenies of competing alignment-free approaches. I implemented *FSWM* in C++ and used the OpenMP library [86] for parallelization. *FSWM* is available as command line tool and through our web server at `http://www.fswm.gobics.de/`.
Furthermore, I investigated if a slightly modified version of *FSWM* can be used as anchor points for genome alignments (see Chapter 3). To do so, I integrated *FSWM* into the popular multiple-genome-alignment pipeline *mugsy* [6] and found out that the anchor points based on *FSWM* outperformed the default anchoring procedure of *mugsy*. When *FSWM* was used to identify anchor points, more homologies were found and aligned and the alignments were of higher quality. Especially for distantly related organisms, a significant improvement was achieved with *FSWM* as anchors.
Moreover, I explored the possibility to transfer the basic *FSWM* idea to protein sequences. Together with a bachelor student, I developed a tool called *Prot-SpaM* (see Chapter 4) which calculates evolutionary distances between whole proteoms within a few seconds. To our knowledge, our tool is the only alignment-free approach that estimates evolutionary distances between protein sequences. Systematic tests on simulated and real-word data showed that the estimated distances approximated the substitution rate very well and also the reconstructed phylogenies based on these distances were of high quality.

5.1 Applications for filtered spaced-word matches

Sequence comparison is a central task in biology. Traditionally, sequence alignments are used to identify similarities between biological sequences. Perhaps the most famous tool to identify local homologies is *BLAST* [3] and its derivatives as for example *PSI-BLAST* [4]. Later, faster high-throughput programs were developed such as *DIAMOND*[12] and *MMseq* [39, 109, 110]. These tools are designed to rapidly align reads to a reference database but they are usually not used to compare pairs of whole genomes. Therefore, I developed the *filtered spaced-word matches* approach which is an efficient and effective tool to identify similarities between sequences without the computationally expensive alignment step. At first, *FSWM* was developed for fast alignment-free whole genome phylogeny reconstruction (see Chapter 2) but later, I identified two more applications where *FSWM* can be used (see Chapter 3 4). In the following three sections, I discuss each application separately.

5.1.1 Whole genome phylogeny reconstruction

Alignment-free sequence comparison has become a vibrant field of research in bioinformatics and a large variety of different approaches were proposed over the past decades. Most traditional alignment-free methods are either based on word counts (see Section 1.1.2) or on match lengths (see Section 1.1.3). Recently, a new idea to estimate distances emerged which is based on micro-alignments (see Section 1.1.4). I discussed the advantages of these methods compared to other alignment-free approaches but also pointed out that current approaches are limited to closely related organisms.
In my research, I focused on the development of a new alignment-free method for phylogeny reconstruction that overcomes the limitation of current approaches (see Chapter 2). Similar to *CO-Phylog* [133] and *andi* [38], *FSWM* is based on micro-alignments. Both competing approaches use a minimal length criteria of the matching word pairs to distinguish between background and homologous hits. This approach, however, becomes increasingly imprecise the longer the sequences are. The relationship between the sequence length and the number of homologous and background matches was pointed out in Section 1.1.2. To alleviate this problem both approaches use additionally a uniqueness criteria, as described in Section 1.1.4. The effectiveness is, however, limited. Instead of filtering matches by length and uniqueness, I proposed a new filtering procedure which calculates a score for

each spaced-word match and discards matches with a score below a threshold. To investigate which threshold should be used, I plotted the number of spaced-word matches against their score. We called these plots *spamograms* (spaced-word match histograms) (see Chapter 4). These *spamograms* showed that a threshold of 0 separates most homologous matches from background matches (see Chapter 2 Figure 1) and therefore I used this as default parameter. If background matches are filtered out, patterns with fewer match positions can be used without losing the signal between the sequences due to noise. This is especially important if the sequences are more distantly related because the number of homologous word matches decrease with increasing sequence divergence but the number of background matches remain the same. Consequently, the *FSWM* approach is competitive with *CO-Phylog* and *andi* for closely related sequences but outperforms both for more distantly related sequences. I showed that *FSWM* is able to estimate substitution rates up to about 0.9 and is less influenced by insertions and deletions then other approaches (see Chapter 2 Figure 2). Moreover, the resulting phylogenies based on distances calculated with *FSWM* were more accurate then the phylogenies based on distances calculated by other alignment-free approaches (see Chapter 2 Figure 3).

One notorious problem associated with whole genome comparison are repetitive regions and gene duplications. For example, a spaced-word can occur at position i in sequence S_1 and the same spaced-word occurs in sequence S_2 at position j and j', i.e. there are two matches, and both matches have a positive score. In this scenario one could assume that only one match occurs due to orthology and the other match occurs due to duplication. For phylogeny reconstruction, the orthologs are used to estimate distances and must be identified first [98, 126, 45]. To this end, I proposed to use a greedy one-to-one mapping of spaced-word matches which works as follows: at first all spaced-word matches with a positive score are stored in a list. Then, the spaced-word match with the highest score is picked. Next, the number of matching and non-matching nucleotides at the *don't care* positions of this spaced-word match is determined and this match and all other matches that start at the same positions in S_1 or S_2 are removed from the list. This procedure is repeated with the next highest-scoring spaced-word match in the list until the list is empty. *CO-Phylog* and *andi*, on the other hand, implicitly deal with duplications by their respective uniqueness criteria, i.e. word matches where no clear assignment is possible are ignored.

The run time of *FSWM* is sufficiently fast to compare large pairs of sequence in reasonable time. For example, it took *FSWM* about one and a half day to calculate all pairwise

distances of a large eukaryotic data, consisting of 14 taxa with a total size of 4.8 gb. In the study presented in Chapter 2, *FSWM* was considerably slower than *andi* but faster than *CO-Phylog* . Complexity-wise, however, *CO-Phylog* should be much faster than *FSWM* but the published implementation leaves room for improvements. The issue of the run time of *FSWM* is discussed in Section 5.2.

Web server

To make the *FSWM* approach accessible to a broad range of researchers, I developed a user-friendly web server. Users can upload a set of sequences in Multi-FASTA format and the number of match positions of the pattern, called the *weight*, can be specified. Then, the job is executed on a powerful server and the user can track the progress of the job via the assigned id. After the calculations are finished the *spamograms* are visualized on

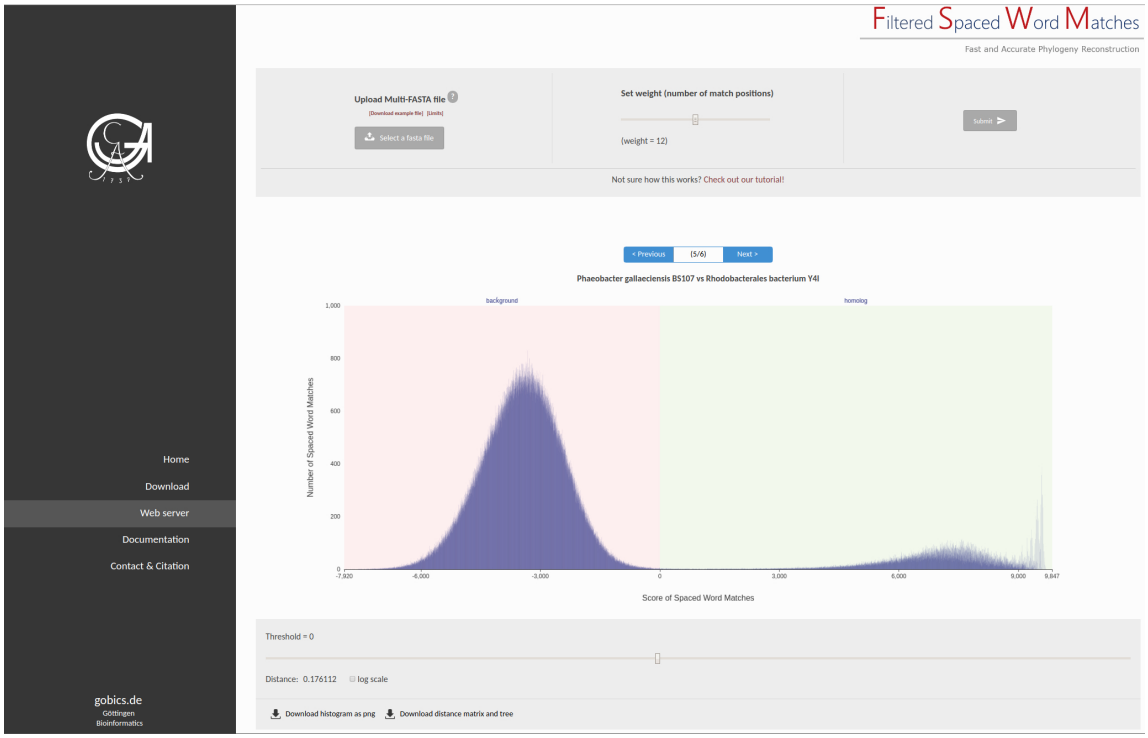

Figure 2: Screenshot of the *FSWM* web server, available at `http://fswm.gobics.de`

the website and the user can switch between them. The visualization of the *spamograms* helps the user to identify a good cut-off value between random and homologous matches if the default threshold does not separate them properly. The threshold can be changed independently for each *spamogram* with the scroll bar at the bottom. If the threshold is changed for one sequence pair, the resulting distance is dynamically recalculated and shown on the website. Once the user is satisfied with the selected thresholds, the distance matrix and corresponding tree, calculated with *Neighbor-Joining*, can be downloaded. Also the *spamograms* can be saved as an image.
There are some restrictions on the web server to prevent server overload. The total file size is limited to 512 mb and the number of sequences is restricted to 100. If large-scale data sets are to be analysed or for researchers with knowledge in linux, it is preferable to use the command line version of *FSWM*. The command line tool, however, does not provide a visualization of the *spamograms*. A solution to this problem could be to upload a smaller subset of sequences on the web server to assess which threshold works best for most sequences pairs.

5.1.2 Anchor points for whole genome alignments

Pairwise or multiple sequence alignments are usually limited to rather short sequences such as single genes or mitochondrial genomes, due to their run time. However, since the late 1990s efforts have been made to align longer sequences. To circumvent the run time problem, it is necessary to reduce the search space to the most promising regions where homologies can be found. Such high-scoring local homologies are usually identified by fast word-based methods and are called *anchor points* [79]. After a sufficient number of local homologies are identified they are ordered and the segments between these anchor points are aligned with a more sensitive but slower alignment method. Perhaps the first approach which was used to identify anchor points for whole genome alignments was *MUMmer* [21, 57]. *MUMmer* rapidly identifies maximal unique matches between pairs of sequences. Another genome alignment program, called *MGA* [43], uses maximal unique matches across all input sequences as anchor points. Both tools work very well on closely related sequences but for more distantly related sequences they reach their limits. This is due to the fact that with increasing sequence divergence the number of maximal unique matches decrease. The performance of genome aligners is strongly dependent on the num-

ber and quality of the anchor points.

In Chapter 3, I showed that *FSWM* can be used to identify high quality anchor points for genome alignments. The idea to use spaced-word matches as anchor points is generally not new and there are already tools available that implemented that approach, e.g. *progressiveMauve* [17] and *cactus* [88]. These tools, however, do not remove spurious random spaced-word matches. Therefore, I proposed to use the filtering technique from *FSWM*. But before *FSWM* can be used as anchor points, two problems must be solved. One issue is that filtered spaced-word matches overlap and another problem is that a spaced-word match can reach into a non-homologous or gapped region. The second problem is solved by performing a gap-free extension of the spaced-word matches from the middle position in both directions instead of extending the matches from the start and end position. This extension also solves the problem of overlapping spaced-word matches because if a spaced-word match is located within an extension, it is redundant and ignored. This procedure is computationally slightly more demanding but still competitive compared to other anchoring procedures.

To evaluate this new anchoring approach, I integrated it into the genome aligner *mugsy* [6]. By default, *mugsy* uses *MUMmer*, i.e. maximal unique matches, to identify anchor points between pairs of genomes. I replaced *MUMmer* with the adjusted version of *FSWM* and left the subsequent alignment procedure untouched. I showed that *FSWM* as anchor points are far superior compared to maximal unique matches if the sequences are distantly related. For very similar sequences both anchoring procedures were equally good. Moreover, *mugsy* with the anchor points from *FSWM* is competitive with *cactus* [88], one of the best genomes aligners available. For some data sets, it even outperformed *cactus* (see Chapter 3 Figures 2-4).

Additionally, I compared the run time of *FSWM* and *MUMmer* with and without the subsequent alignment procedure (see Chapter 3 Table 2). The run time of *FSWM* as standalone program is comparable to *MUMmer* but the run time of the subsequent alignment pipeline differs substantially. If *FSWM* is used to identify anchor points, the run time of the alignment procedure is much higher. This is primarily due to the fact that more anchor points are found by *FSWM* and thus more regions must be aligned. In conclusion, *FSWM* is a good alternative to standard anchoring procedures. The resulting alignments are of very high quality and the run time is within an acceptable range.

5.1.3 *Prot-SpaM:* whole proteome phylogeny reconstruction

In Chapter 4, I introduced the *Proteome-based Spaced-Word Matches (Prot-SpaM)* approach. I transferred the idea of *FSWM* from genomic sequences to protein sequences. Using protein sequences for phylogeny reconstruction has some advantages over nucleotide sequences. For example, protein sequences are more conserved than DNA sequences. Homologies between distal protein sequences can be still found even if they are almost not detectable anymore on the DNA level. Therefore, it is desirable to have alignment-free methods for protein sequences. Current alignment-free methods for protein comparison are not based on a stochastic model of evolution. *Prot-SpaM*, on the other hand, is able to estimates the number of substitutions between protein sequences. Similar to *FSWM* for DNA sequences, *Prot-SpaM* searches for local gap-free alignments from which the distances are estimated. To calculate an evolutionary distance we used the *Kimura* model [54] which approximates the *PAM* distance [20]. Spurious random matches are identified based on a score which is calculated with the *BLOSUM62* substitution matrix [40] based on the aligned amino acids at the *don't care* positions.
The *Prot-SpaM* distances were evaluated based on simulated data (see Chapter 4 Figure 2-3) while the phylogeny analysis was performed on real-word proteoms (see Chapter 4 Figures 4-7 and Table 2). The correct phylogeny of real-world organisms is often unknown. To circumvent this issue, we worked together with experts for some data sets to assess the quality of the resulting phylogenies. I showed that the estimated distances reflect the number of substitutions very accurately up to 2.0 substitutions per site. Moreover, for most data sets, the resulting phylogenies were better compared to the trees based on other alignment-free approaches. The run time of *Prot-SpaM* is much lower compared to the nucleotide version of *FSWM*. This is due to the fact that the sequence length of protein sequences is substantially shorter than DNA sequences. Additionally, no reverse complement of the sequences must be taken into account. *Prot-SpaM* is usually faster compared to other alignment-free approaches, except for *FFP* [49] (see Chapter 4 Table 3).

5.2 Limitations

I identified three different applications where *FSWM* can be used (see Chapter 2 3 4). In most cases, *FSWM* outperformed current state-of-the art methods. There are, however, limitations which I discuss in this section.

As described in the introduction (see Chapter 1.1.2), it is a challenge to distinguish signal from noise between pairs of sequences. Instead of filtering word-matches by their lengths, I proposed to calculate a score for each spaced-word match and then discard those which have a score below a threshold. As I demonstrated in chapter 2, the results of *FSWM* are very good but the score calculation is computationally expensive. The problem is that if a certain spaced-word occurs n times in sequence S_1 and m times in sequence S_2 then there are $n \times m$ matches that involve this spaced-word and $n \times m$ calculations must be performed. Consequently, the run time is dependent on the number of matches that involve occurrences of the same spaced-word. Therefore, the distribution of the spaced-words must be investigated to evaluate the complexity of *FSWM*. For two sequences with independent and identically distributed characters, the spaced-words are equally distributed, i.e. each spaced-word occurs with about the same frequency in both sequences. This is computationally the best case scenarios because the score calculation of the matches which involve a certain spaced-word takes time proportional to the product of the occurrences of this spaced-word in S_1 and S_2. The worst case scenario, on the other hand, would be if two sequences of length L consist of repetitions of the same letter. In this case, $\mathcal{O}(L^2)$ computations must be executed which takes as much time as to calculate an optimal alignment with the *Needleman-Wunsch* algorithm [84]. Empirically, most real-world sequences are somewhere between these two extremes and I demonstrated that *FSWM* can calculate distances between pairs of eukaryotic genomes of a few hundred mb within a few minutes. However, such a low run time can not be guaranteed for all data sets. Especially high repetitive segments can cause a steep increase in the run time because many matches are found in such regions. Such regions could be marked and removes from further downstream analysis. However, no such tool is integrated in *FSWM* yet.

The run time of *FSWM* also depends on the number of match positions of the underlying pattern. The smaller the number of match positions, the more matches are found and the more time is required to calculate the score between all matches. Tests showed that the accuracy of the distances are not systematically influenced by the number of match

positions. Therefore, it is preferable to use patterns with more match positions for larger sequences because it reduces the run time significantly.
Another issue that needs to be discussed is that *FSWM*, by default, uses patterns with 100 *don't care* and 12 *match* positions. Such long patterns are necessary to clearly separate homologous from background matches. The disadvantage is that homologies are missed which contain insertions or deletions. Missing spaced-word matches is generally not an issue as long as a sufficient number of matches are found such that the distances can be still estimated properly. It is more problematic if spaced-word matches are found that are partially homologous and partially random but still have a score above the threshold. In such cases, a higher mismatch rate is calculated from the spaced-word matches and the resulting distances are overestimated. Empirically, such incidences are fairly low and do not have a large impact on the distances. Therefore, I ignored them. For the anchoring approach (see Chapter 3), I solved this issue by extending the spaced-word matches from their midpoint with a standard X-drop approach. This procedure, however, increases the run time and therefore I did not use it for the distance estimation for phylogeny reconstruction.
FSWM can precisely estimate distances up to 0.9 substitutions per site, then the curve begins to flatten out (see Chapter2 Figure 2). This is due to the fact that the score of the homologous matches decreases with increasing substitution rate. At about 0.9 to 1.0 substitutions per site homologous matches and background matches become indistinguishable by the score and the distance accuracy drops rapidly. If more than 100 *don't care* positions are used, distances between more distantly related sequences can be estimated but the underlying sequences must have longer regions without insertion or deletions. If less than 100 *don't care* positions are used more homologous matches can be found in gapped regions but only smaller substitution rates can be estimated. Empirically, 100 *don't care* positions work well for most data sets but for some data, the default parameters might need to be adjusted.
The patterns in *FSWM* are generated and optimized with *rasbhari* [32] to ensure a lower statistical dependency between overlapping spaced-words. The optimization procedure in *rasbhari* is not deterministic and the results of *FSWM* vary slightly in each program run. This is a clear disadvantage. The alternative to set a fixed pattern is also problematic because there is no single pattern that works best for all data sets. For the nucleotide version of *FSWM* the influence of the pattern is very low and in most cases negligible. Also the resulting phylogenies are very stable for different patterns. In contrast, *Prot-SpaM* is

strongly dependent on the underlying pattern. The distances and the resulting phylogenies vary significantly for different patterns. In order to reduce the variance, we use sets of multiple patterns. The effect that multiple patterns lead to much better and more stable results for protein sequences was already shown in a previous study [60]. However, even if multiple patterns are used, the variance of *Prot-SpaM* still remains higher compared to the nucleotide version.

In the last part of this section, I discuss the general evaluation procedure of alignment-free methods and point out their weaknesses. Alignment-free methods are usually evaluated based on simulated sequence data as well as on real-world sequences. The advantage of simulated data is that the evolutionary truth is known which makes benchmarking easy. The disadvantage is that sequence simulators usually use simplified models of evolution which can not cover the whole spectrum of evolutionary events. Consequently, the assessment of the results of simulated data sets is limited. For real-word sequences, on the other hand, the correct phylogeny is often unknown or there are even disagreements between research groups. Therefore, developers of alignment-free methods often rely on published phylogenies that are often inferred based on selected marker genes or sometimes even based on 16S rRNA. The quality of these trees are often unknown and therefore it is often problematic to evaluate the performance of alignment-free methods. In our *Prot-SpaM* paper, we tried to circumvent this problem by working closely together with experts to evaluate the resulting phylogenies. A good solution would be standard benchmark data sets where experts provide reliable phylogenies such that the accuracy of alignment-free methods can be investigated properly.

6 Outlook

The outlook section is divided into two parts. In the first part, I describe ideas how *FSWM* could be improved and in the second section I suggest other possible application for *FSWM*.

6.1 Possible improvements

I showed that *FSWM* estimates substitution rates between unaligned genomes more accurately than other alignment-free methods. This accuracy, however, comes at a price: the run time is in most cases higher compared to competing approaches. Given the fact, that in the future much more biological data will be produced [111] the biggest improvement of *FSWM* would be a reduction of the run time. The bottleneck of *FSWM* is the exhaustive score calculation of each spaced-word match. This problem could be overcome if not scores of all the spaced-word matches are calculated but only of a fraction of them, i.e. some spaced-word matches are sampled. This strategy would not affect the estimated distances because the distances are calculated based on the aligned nucleotides at the *don't care* positions of the spaced-word matches and not based on the *number* of spaced-word matches. For the anchoring application of *FSWM*, on the other hand, some homologies could be missed if spaced-word matches are sampled. But this effect might be limited because of the extension of the spaced-word matches. It would be sufficient if one single match is found within a homologous region.

In the following, I describe three different sampling strategies that could be used to decrease the run time. The first sampling strategy works as follows: one spaced-word from sequence S_1 is randomly picked and compared the same spaced-word occurring somewhere in sequence S_2, i.e. the score of this spaced-word match is calculated. If this match has a positive score, the fraction of non-matching nucleotides (or amino acids) at the *don't care* positions are determined and this spaced-word match is deleted. If the score is negative, another position in S_2 is randomly picked where the same spaced-word occurs and the score is calculated again. This procedure is repeated either until a match with a positive score is found, a user defined limit of comparisons is reached or until the scores for all spaced-word matches are calculated. Then, the next spaced-word from S_1 is picked, which can either be the same spaced-word occurring at another position or a different spaced-word. Once a sufficient number of spaced-word matches are sampled, i.e. until the estimated distances are stable, the algorithm stops. This strategy is similar to the sampling procedure in *Multi-Spam* [22].

The second sampling strategy does not randomly select spaced-words to calculate a score but instead spaced-word matches are picked that have a higher chance for a positive score. To this end, all equal spaced-words from S_1 and S_2 are determined and sorted together

in one list according to their *don't care* positions. Then, only the scores of neighboring spaced-words in this list are calculated. The chance of a positive score for neighboring spaced-words is higher because equal nucleotides at the *don't care* positions are next to each other in the sorted list. A similar idea was implemented in the *UProC* approach for ultra-fast protein domain classification [78].
The third and last sampling idea is to sample spaced-words from the sequences directly. That means not all spaced-words occurring in a sequence are determined but only a predefined fraction. One simple rule could be, for example, that only spaced-words from even positions are used. On top of this sampling, the other two strategies described above could be used additionally.
Besides the run time, there are two more ideas how *FSWM* for whole genome phylogeny reconstruction could be developed further. The first idea is to use another substitution model than Jukes&Cantor [48] to calculate substitution rates. For example, the *Kimura two parameter model* [53] could be used because the fraction of transversions and transitions can be determined easily by comparing the aligned nucleotides at the *don't care* positions. This is, however, a very fine-grained tuning whose effect might be limited. A more promising improvement could be to calculate distances only based on so called singleton matches. These are spaced-words that only match one position in each sequence with a positive score. This would be an alternative to the one-to-one mapping and the distances may be less influenced by repeats. However, for all of these ideas new studies must be conducted to evaluate which strategies yield the best results.

6.2 Further applications

In this section, I describe two other potential applications for *FSWM*. Perhaps one of the most popular topic in bioinformatics is metagenomic classification, see [11] for a recent review. One established tool for taxonomic classification is called *Kraken* [129]. It rapidly identifies all reads in a metagenomic sample by comparing them to a database of known genomes. To do so, *Kraken* uses long exact word matches of a fixed length instead of computationally more expensive alignments. If one read matches to multiple genomes in the database, then the read is assigned to the lowest common ancestor of those taxa. Therefore, it is important to avoid random matches which they achieve by using word matches of length 31. The disadvantage is that only matches between closely related

organisms can be found. This problem could be solved if *FSWM* is used instead, because it can distinguish between background and homologous hits. There are, however, some problems that must be addressed first. One issue is that reads are short and *FSWM* by default uses patterns of length 112. Shorter patterns seem to be preferable but it needs to be explored how much the length can be reduced until homologous matches cannot be distinguished from background matches anymore. Another problem is that the run time might be too high for very large metagenomic samples. This could be solved by the sampling strategies proposed above. The same two problems also apply if distances between two organisms are to be calculated based on unassembled reads.

Another application of *FSWM* could be the identification of regions where horizontal gene transfers happened. The homologous peak of the *spamograms* reflects different degrees of sequence similarity in different regions of the sequences (e.g. see Chapter 2 Figure 1D). Regions with a considerably higher score compared to the rest of the genome could be an indication of horizontal gene transfer. With *FSWM* such regions could be very quickly identified and further investigated with either slower but more precise methods or manually by experts.

In this thesis, I conducted three studies in which *FSWM* could be applied with great effect and I described two more possible application in this section. This highlights the generality and versatility of *FSWM* which could serve as foundation of further alignment-free research and development.

References

[1] Mohamed Ibrahim Abouelhoda, Stefan Kurtz, and Enno Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2:53 – 86, 2004.

[2] Athena Ahmadi, Alexander Behm, Nagesh Honnalli, Chen Li, Lingjie Weng, and Xiaohui Xie. Hobbes: optimized gram-based methods for efficient read alignment. *Nucleic Acids Research*, 40:e41, 2011.

[3] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene M. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[4] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25: 3389–3402, 1997.

[5] Sasha K. Ames, David A. Hysom, Shea N. Gardner, G. Scott Lloyd, Maya B. Gokhale, and Jonathan E. Allen. Scalable metagenomic taxonomy classification using a reference genome database. *Bioinformatics*, 29(18):2253–2260, 2013.

[6] Samuel V. Angiuoli and Steven L. Salzberg. Mugsy: fast multiple alignment of closely related whole genomes. *Bioinformatics*, 27:334–342, 2011.

[7] Guillaume Bernard, Cheong Xin Chan, and Mark A Ragan. Alignment-free microbial phylogenomics under scenarios of sequence divergence, genome rearrangement and lateral genetic transfer. *Scientific Reports*, 6:28970, 2016.

[8] B E Blaisdell. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*, 83(14):5155–5159, 1986.

[9] Mark S. Boguski. Bioinformatics - a new era. *Trends in Biotechnology*, 16:1–3, 1998.

[10] Oliver Bonham-Carter, Joe Steele, and Dhundy Bastola. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. *Briefings in Bioinformatics*, 15:890–905, 2014.

[11] Florian P. Breitwieser, Jennifer Lu, and Steven L. Salzberg. A review of methods and databases for metagenomic classification and assembly. *Briefings in Bioinformatics*, 2017. doi: 10.1093/bib/bbx120.

[12] Benjamin Buchfink, Chao Xie, and D. H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12:59–60, 2015.

[13] Sourav Chatterji, Ichitaro Yamazaki, Zhaojun Bai, and Jonathan A. Eisen. Compostbin: A dna composition-based algorithm for binning environmental shotgun reads. In Martin Vingron and Limsoon Wong, editors, *RECOMB*, volume 4955 of *Lecture Notes in Computer Science*, pages 17–28. Springer, 2008.

[14] Matteo Comin and Davide Verzotto. The irredundant class method for remote homology detection of protein sequences. *Journal of Computational Biology*, 18:1819–1829, 2011.

[15] Matteo Comin and Davide Verzotto. Alignment-free phylogeny of whole genomes using underlying subwords. *Algorithms for Molecular Biology*, 7:34, 2012.

[16] Torney D., Burks C., Davison D., and Sirotkin K. Computation of d^2 : a measure of sequence dissimilarity. *Computers and DNA: Proc. Interfac. Comput. Sci. Nucleic Acid Seq. Workshop, Santa Fe, N.M.*, 1990.

[17] Aaron E. Darling, Bob Mau, and Nicole T. Perna. progressivemauve: Multiple genome alignment with gene gain, loss and rearrangement. *PLOS ONE*, 5(6):1–17, 2010.

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

[19] M. Dayhoff, R. Schwartz, and B. Orcutt. A model of evolutionary change in proteins. In M. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, pages 345–352. National Biomedical Research Foundation, 1978.

[20] M.D. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 6:345–362, 1978.

[21] Arthur L. Delcher, Simon Kasif, Robert D. Fleischmann, Jeremy Peterson, Owen White, and Steven L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27:2369–2376, 1999.

[22] Thomas Dencker, Chris-André Leimeister, Michael Gerth, Christoph Bleidorn, Sagi Snir, and Burkhard Morgenstern. Multi-SpaM: a Maximum-Likelihood approach to phylogeny reconstruction using multiple spaced-word matches and quartet trees. In Mathieu Blanchette and Aïda Ouangraoua, editors, *Comparative Genomics*, pages 227–241. Springer International Publishing, 2018.

[23] Alexandre Drouin, Sébastien Giguère, Maxime Déraspe, Mario Marchand, Michael Tyers, Vivian G. Loo, Anne-Marie Bourgault, François Laviolette, and Jacques Corbeil. Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons. *BMC Genomics*, 17:754, 2016.

[24] Maria Federico, Mauro Leoncini, Manuela Montangero, and Paolo Valente. Direct vs 2-stage approaches to structured motif finding. *Algorithms for Molecular Biology*, 7:20, 2012.

[25] Joseph Felsenstein. Evolutionary trees from dna sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.

[26] Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA, USA, 2004.

[27] Johannes H Fischer and Volker Heun. A new succinct representation of rmq-information and improvements in the enhanced suffix array. In *ESCAPE*, 2007.

[28] Walter M. Fitch. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20(4):406–416, 1971.

[29] GE Fox, E Stackebrandt, RB Hespell, J Gibson, J Maniloff, TA Dyer, RS Wolfe, WE Balch, RS Tanner, LJ Magrum, LB Zablen, R Blakemore, R Gupta, L Bonen, BJ Lewis, DA Stahl, KR Luehrsen, KN Chen, and CR Woese. The phylogeny of prokaryotes. *Science*, 209(4455):457–463, 1980.

[30] Lei Gao, Ji Qi, JianDong Sun, and BaiLin Hao. Prokaryote phylogeny meets taxonomy: An exhaustive comparison of composition vector trees with systematic bacteriology. *Science in China Series C: Life Sciences*, 50(5):587–599, 2007.

[31] Dan Gusfield and Jens Stoye. Linear time algorithms for finding and representing all the tandem repeats in a string. *Journal of Computer and System Sciences*, 69(4): 525 – 546, 2004.

[32] Lars Hahn, Chris-Andre Leimeister, Rachid Ounit, Stefano Lonardi, and Burkhard Morgenstern. rasbhari: Optimizing spaced seeds for database searching, read mapping and alignment-free sequence comparison. *PLOS Computational Biology*, 12(10): 1–18, 2016.

[33] BAILIN HAO and JI QI. Prokaryote phylogeny without sequence alignment: From avoidance signature to composition distance. *Journal of Bioinformatics and Computational Biology*, 02(01):1–19, 2004.

[34] Masami Hasegawa, Hirohisa Kishino, and Taka-aki Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of Molecular Evolution*, 22(2):160–174, 1985.

[35] Bernhard Haubold. Alignment-free phylogenetics and population genetics. *Briefings in Bioinformatics*, 15:407–418, 2014.

[36] Bernhard Haubold, Nora Pierstorff, Friedrich Möller, and Thomas Wiehe. Genome comparison without alignment using shortest unique substrings. *BMC Bioinformatics*, 6(1):123, 2005.

[37] Bernhard Haubold, Peter Pfaffelhuber, Mirjana Domazet-Loso, and Thomas Wiehe. Estimating mutation distances from unaligned genomes. *Journal of Computational Biology*, 16:1487–1500, 2009.

[38] Bernhard Haubold, Fabian Klötzl, and Peter Pfaffelhuber. andi: Fast and accurate estimation of evolutionary distances between closely related genomes. *Bioinformatics*, 31:1169–1175, 2015.

[39] Maria Hauser, Martin Steinegger, and Johannes Söding. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics*, 32:1323–1330, 2016.

[40] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.

[41] Cody E. Hinchliff, Stephen A. Smith, James F. Allman, J. Gordon Burleigh, Ruchi Chaudhary, Lyndon M. Coghill, Keith A. Crandall, Jiabin Deng, Bryan T. Drew, Romina Gazis, Karl Gude, David S. Hibbett, Laura A. Katz, H. Dail Laughinghouse, Emily Jane McTavish, Peter E. Midford, Christopher L. Owen, Richard H. Ree, Jonathan A. Rees, Douglas E. Soltis, Tiffani Williams, and Karen A. Cranston. Synthesis of phylogeny and taxonomy into a comprehensive tree of life. *Proceedings of the National Academy of Sciences*, 112(41):12764–12769, 2015.

[42] Michael Höhl and Mark A. Ragan. Is multiple-sequence alignment required for accurate inference of phylogeny? *Systematic Biology*, 56(2):206–221, 2007.

[43] Michael Höhl, Stefan Kurtz, and Enno Ohlebusch. Efficient multiple genome alignment. *Bioinformatics*, 18:312S–320S, 2002.

[44] Sebastian Horwege, Sebastian Lindner, Marcus Boden, Klaus Hatje, Martin Kollmar, Chris-André Leimeister, and Burkhard Morgenstern. *Spaced words* and *kmacs*: fast alignment-free sequence comparison based on inexact word matches. *Nucleic Acids Research*, 42(W1):W7–W11, 2014.

[45] Jaime Huerta-Cepas, Damian Szklarczyk, Kristoffer Forslund, Helen Cook, Davide Heller, Mathias C. Walter, Thomas Rattei, Daniel R. Mende, Shinichi Sunagawa, Michael Kuhn, Lars Juhl Jensen, Christian von Mering, and Peer Bork. eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic Acids Research*, 44:D286–D293, 2016.

[46] Lucian Ilie and Silvana Ilie. Multiple spaced seeds for homology search. *Bioinformatics*, 23:2969–2977, 2007.

[47] Lucian Ilie, Silvana Ilie, and Anahita M. Bigvand. SpEED: fast computation of sensitive spaced seeds. *Bioinformatics*, 27:2433–2434, 2011.

[48] Thomas H. Jukes and Charles R. Cantor. *Evolution of Protein Molecules*. Academy Press, New York, 1969.

[49] Se-Ran Jun, Gregory E. Sims, Guohong A. Wu, and Sung-Hou Kim. Whole-proteome phylogeny of prokaryotes by feature frequency profiles: An alignment-free method

with optimal feature resolution. *Proceedings of the National Academy of Sciences*, 107(1):133–138, 2010.

[50] Miriam Kantorovitz, Gene Robinson, and Saurabh Sinha. A statistical method for alignment-free comparison of regulatory sequences. *Bioinformatics*, 23:249–255, 2007.

[51] Samuel Kariin and Chris Burge. Dinucleotide relative abundance extremes: a genomic signature. *Trends in Genetics*, 11(7):283 – 290, 1995.

[52] Samuel Karlin and Jan Mrázek. Compositional differences within and between eukaryotic genomes. *Proceedings of the National Academy of Sciences*, 94(19):10227–10232, 1997.

[53] Motoo Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, 1980.

[54] Motoo Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1984.

[55] Stefan Kurtz. Reducing the space requirement of suffix trees. *Softw. Pract. Exper.*, 29(13):1149–1171, 1999.

[56] Stefan Kurtz, Adam Phillippy, Arthur L. Delcher, Michael Smoot, Martin Shumway, Corina Antonescu, and Steven L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biology*, 5(2), 2004.

[57] Stefan Kurtz, Adam Phillippy, Arthur L. Delcher, Michael Smoot, Martin Shumway, Corina Antonescu, and Steven L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biology*, 5:R12+, 2004.

[58] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10:R25, 2009.

[59] Chris-André Leimeister and Burkhard Morgenstern. *kmacs*: the k-mismatch average common substring approach to alignment-free sequence comparison. *Bioinformatics*, 30:2000–2008, 2014.

[60] Chris-André Leimeister, Marcus Boden, Sebastian Horwege, Sebastian Lindner, and Burkhard Morgenstern. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics*, 30:1991–1999, 2014.

[61] Chris-André Leimeister, Salma Sohrabi-Jahromi, and Burkhard Morgenstern. Fast and accurate phylogeny reconstruction using filtered spaced-word matches. *Bioinformatics*, 33:971–979, 2017.

[62] Chris-Andre Leimeister, Thomas Dencker, and Burkhard Morgenstern. Accurate multiple alignment of distantly related genome sequences using filtered spaced word matches as anchor points. *Bioinformatics*, 2018. doi: 10.1093/bib/bbx120. in press.

[63] Chris-Andre Leimeister, Jendrik Schellhorn, Svenja Schoebel, Michael Gerth, Christoph Bleidorn, and Burkhard Morgenstern. Prot-spam: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *GigaScience*, 2018. to appear.

[64] Christina S. Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for svm protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575, Singapore, 2002. World Scientific Publishing.

[65] Garmay Leung and Michael B. Eisen. Identifying cis-regulatory sequences by word profile similarity. *PLOS ONE*, 4(9):1–11, 2009.

[66] Henry C. M. Leung, S. M. Yiu, Bin Yang, Yu Peng, Yi Wang, Zhihua Liu, Jingchi Chen, Junjie Qin, Ruiqiang Li, and Francis Y. L. Chin. A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. *Bioinformatics*, 27:1489–1495, 2011.

[67] Ming Li, Bin Ma, Derek Kisman, and John Tromp. PatternHunter II: Highly sensitive and fast homology search. *Genome Informatics*, 14:164–175, 2003.

[68] Ming Li, Bin Ma, Derek Kisman, and John Tromp. PatternHunter II: highly sensitive and fast homology search. *Journal of Bioinformatics and Computational Biology*, 02: 417–439, 2004.

[69] Qiang Li, Zhao Xu, and Bailin Hao. Composition vector approach to whole-genome-based prokaryotic phylogeny: Success and foundations. *Journal of Biotechnology*, 149(3):115 – 119, 2010.

[70] Ruiqiang Li, Yingrui Li, Karsten Kristiansen, and Jun Wang. SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24:713–714, 2008.

[71] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.

[72] T. Lingner and P. Meinicke. Remote homology detection based on oligomer distances. *Bioinformatics*, 22:2224–2231, 2006.

[73] Thomas Lingner and Peter Meinicke. Word correlation matrices for protein sequence analysis and remote homology detection. *BMC Bioinformatics*, 9:259, 2008.

[74] Ross A. Lippert, Haiyan Huang, and Michael S. Waterman. Distributional regimes for the number of k-word matches between two random sequences. *Proceedings of the National Academy of Sciences*, 99(22):13980–13989, 2002.

[75] Yang Young Lu, Kujin Tang, Jie Ren, Jed A. Fuhrman, Michael S. Waterman, and Fengzhu Sun. Cafe: accelerated alignment-free sequence analysis. *Nucleic Acids Research*, 45(W1):W554–W559, 2017.

[76] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, SODA '90:319–327, 1990.

[77] Guillaume MarÃğais, Arthur L. Delcher, Adam M. Phillippy, Rachel Coston, Steven L. Salzberg, and Aleksey Zimin. Mummer4: A fast and versatile genome alignment system. *PLOS Computational Biology*, 14(1):1–14, 2018.

[78] Peter Meinicke. UProC: tools for ultra-fast protein domain classification. *Bioinformatics*, 31:1382–1388, 2015.

[79] Burkhard Morgenstern, Sonja J. Prohaska, Dirk Pöhler, and Peter F. Stadler. Multiple sequence alignment with user-defined anchor points. *Algorithms for Molecular Biology*, 1:6, 2006.

[80] Burkhard Morgenstern, Bingyao Zhu, Sebastian Horwege, and Chris-André Leimeister. Estimating evolutionary distances from spaced-word matches. In Dan Brown and Burkhard Morgenstern, editors, *Algorithms in Bioinformatics*, volume 8701 of *Lecture Notes in Computer Science*, pages 161–173. Springer, Berlin Heidelberg, 2014.

[81] Burkhard Morgenstern, Bingyao Zhu, Sebastian Horwege, and Chris-André Leimeister. Estimating evolutionary distances between genomic sequences from spaced-word matches. *Algorithms for Molecular Biology*, 10:5, 2015.

[82] Burkhard Morgenstern, Svenja Schöbel, and Chris-André Leimeister. Phylogeny reconstruction based on the length distribution of k-mismatch common substrings. *Algorithms for Molecular Biology*, 12(1):27, 2017.

[83] Joseph H. Nadeau and David Sankoff. Counting on comparative maps. *Trends in Genetics*, 14(12):495 – 501, 1998.

[84] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48: 443–453, 1970.

[85] Ge Nong. Practical Linear-time O(1)-workspace Suffix Sorting for Constant Alphabets. *ACM Trans. Inf. Syst.*, 31(3), 2013.

[86] OpenMP Architecture Review Board. OpenMP application program interface version 3.0, May 2008. URL `http://www.openmp.org/mp-documents/spec30.pdf`.

[87] Hasan H. Otu and Khalid Sayood. A new sequence distance measure for phylogenetic tree construction. *Bioinformatics*, 19(16):2122–2130, 2003.

[88] Benedict Paten, Dent Earl, Ngan Nguyen, Mark Diekhans, Daniel Zerbino, and David Haussler. Cactus: Algorithms for genome multiple sequence alignment. *Genome Research*, 21:1512–1528, 2011.

[89] Rob Patro, Stephen M. Mount, and Carl Kingsford. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology*, 32:462–464, 2014.

[90] Ji Qi, Hong Luo, and Bailin Hao. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. *Nucleic Acids Research*, 32(suppl 2):W45–W47, 2004.

[91] Ji Qi, Bin Wang, and Bai-Iin Hao. Whole proteome prokaryote phylogeny without sequence alignment: A k-string composition approach. *Journal of Molecular Evolution*, 58(1):1–11, 2004.

[92] Mark A Ragan, Guillaume Bernard, and Cheong Xin Chan. Molecular phylogenetics before sequences. *RNA Biology*, 11(3):176–185, 2014.

[93] Gesine Reinert, David Chew, Fengzhu Sun, and Michael S. Waterman. Alignment-free sequence comparison (i): Statistics and power. *Journal of Computational Biology*, 16(12):1615–1634, 2009.

[94] Jie Ren, Xin Bai, Yang Young Lu, Kujin Tang, Ying Wang, Gesine Reinert, and Fengzhu Sun. Alignment-free sequence analysis and applications. *Annual Review of Biomedical Data Science*, 1(1):93–114, 2018.

[95] Fredrik Ronquist and John P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19:1572–1574, 2003.

[96] David J. Russell, Samuel F. Way, Andrew K. Benson, and Khalid Sayood. A grammar-based distance metric enables fast and accurate clustering of large sets of 16s sequences. *BMC Bioinformatics*, 11(1):601, 2010.

[97] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[98] F Schreiber, G Wörheide, and B Morgenstern. Orthoselect: a web server for selecting orthologous gene alignments from est sequences. *Nucl Acids Res*, 37:W185–188, 2009.

[99] Isabel Schwende and Tuan D. Pham. Pattern recognition and probabilistic measures in alignment-free sequence analysis. *Briefings in Bioinformatics*, 15:354–368, 2014.

[100] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[101] Anish Man Singh Shrestha, Martin C. Frith, and Paul Horton. A bioinformatician's guide to the forefront of suffix array construction algorithms. *Briefings in Bioinformatics*, 15(2):138–154, 2014.

[102] Gregory E. Sims and Sung-Hou Kim. Whole-genome phylogeny of escherichia coli/shigella group by feature frequency profiles (ffps). *Proceedings of the National Academy of Sciences*, 108(20):8329–8334, 2011.

[103] Gregory E. Sims, Se-Ran Jun, Guohong A. Wu, and Sung-Hou Kim. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106:2677–2682, 2009.

[104] Sagi Snir and Satish Rao. Quartet MaxCut: A fast algorithm for amalgamating quartet trees. *Molecular Phylogenetics and Evolution*, 62:1 – 8, 2012.

[105] James M. Sobieski, Kwang Nan Chen, Jay C. Filiatreau, Mark H. Pickett, and George E. Fox. 16s rrna oligonucleotide catalog data base. *Nucleic Acids Research*, 12(1Part1):141–148, 1984.

[106] Robert R Sokal and Charles D Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[107] Kai Song, Jie Ren, Gesine Reinert, Minghua Deng, Michael S. Waterman, and Fengzhu Sun. New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Briefings in Bioinformatics*, 15(3):343–353, 2014.

[108] Alexandros Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30:1312–1313, 2014.

[109] Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

[110] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):2542, 2018.

[111] Zachary D. Stephens, Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, and Gene E. Robinson. Big data: Astronomical or genomical? *PLOS Biology*, 13(7): 1–11, 07 2015.

[112] Frank J. Sulloway. Darwin and his finches: The evolution of a legend. *Journal of the History of Biology*, 15(1):1–53, 1982.

[113] JianDong Sun, Zhao Xu, and BaiLin Hao. Whole-genome based archaea phylogeny and taxonomy: A composition vector approach. *Chinese Science Bulletin*, 55(22): 2323–2328, 2010.

[114] Mingfeng Tan, Dong Yu, Yuan Jin, Lei Dou, Beiping LI, Yuelan Wang, Junjie Yue, and Long Liang. An information transmission model for transcription factor binding at regulatory dna sites. *Theoretical Biology and Medical Modelling*, 9(1):19, 2012.

[115] Olga Tanaseichuk, James Borneman, and Tao Jiang. Separating metagenomic short reads into genomes via clustering. *Algorithms for Molecular Biology*, 7:27, 2012.

[116] S. Tavaré. *Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences*, volume 17, pages 57–86. American Mathematical Society, 1986.

[117] Hanno Teeling, Jost Waldmann, Thierry Lombardot, Margarete Bauer, and Frank Glockner. Tetra: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in dna sequences. *BMC Bioinformatics*, 5:163, 2004.

[118] Hervé Tettelin, Vega Masignani, Michael J. Cieslewicz, Claudio Donati, Duccio Medini, Naomi L. Ward, Samuel V. Angiuoli, Jonathan Crabtree, Amanda L. Jones, A. Scott Durkin, Robert T. DeBoy, Tanja M. Davidsen, Marirosa Mora, Maria Scarselli, Immaculada Margarit y Ros, Jeremy D. Peterson, Christopher R. Hauser, Jaideep P. Sundaram, William C. Nelson, Ramana Madupu, Lauren M. Brinkac, Robert J. Dodson, Mary J. Rosovitz, Steven A. Sullivan, Sean C. Daugherty, Daniel H. Haft, Jeremy Selengut, Michelle L. Gwinn, Liwei Zhou, Nikhat Zafar, Hoda Khouri, Diana Radune, George Dimitrov, Kisha Watkins, Kevin J. B. O'Connor, Shannon Smith, Teresa R. Utterback, Owen White, Craig E. Rubens, Guido Grandi, Lawrence C. Madoff, Dennis L. Kasper, John L. Telford, Michael R. Wessels, Rino Rappuoli, and Claire M. Fraser. Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: Implications for the microbial "pan-genome". *Proceedings of the National Academy of Sciences*, 102(39):13950–13955, 2005.

[119] Douglas L. Theobald. A formal test of the theory of universal common ancestry. *Nature*, 465:219–222, 2010.

[120] Igor Ulitsky, David Burstein, Tamir Tuller, and Benny Chor. The average common substring approach to phylogenomic reconstruction. *Journal of Computational Biology*, 13:336–350, 2006.

[121] Susana Vinga. Information theory applications for biological sequence analysis. *Briefings in Bioinformatics*, 15(3):376–389, 2014.

[122] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison - a review. *Bioinformatics*, 19:513–523, 2003.

[123] Lin Wan, Gesine Reinert, Fengzhu Sun, and Michael S. Waterman. Alignment-free sequence comparison (ii): Theoretical power of comparison statistics. *Journal of Computational Biology*, 17(11):1467–1490, 2010.

[124] LUSHENG WANG and TAO JIANG. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

[125] Yi Wang, Henry C.M. Leung, S.M. Yiu, and Francis Y.L. Chin. MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics*, 28:i356–i362, 2012.

[126] Robert M. Waterhouse, Fredrik Tegenfeldt, Jia Li, Evgeny M. Zdobnov, and Evgenia V. Kriventseva. Orthodb: a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Research*, 41:D358–D365, 2013.

[127] P Weiner. Linear pattern matching algorithms. In *Proc. of the 14th IEEE Symp. on Switching and Automata Theory*, pages 1–11, 1973.

[128] C R Woese. Bacterial evolution. *Microbiology and Molecular Biology Reviews*, 51(2): 221–271, 1987.

[129] Derrick E. Wood and Steven L. Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, Mar 2014.

[130] Guohong Albert Wu, Se-Ran Jun, Gregory E. Sims, and Sung-Hou Kim. Whole-proteome phylogeny of large dsdna virus families by an alignment-free method. *Proceedings of the National Academy of Sciences*, 106(31):12826–12831, 2009.

[131] Yu-Wei Wu and Yuzhen Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *Journal of Computational Biology*, 18:523–534, 2011.

[132] Kuan Yang and Liqing Zhang. Performance comparison between k -tuple distance and four model-based distances in phylogenetic tree reconstruction. *Nucleic Acids Research*, 36(5):e33, 2008.

[133] Huiguang Yi and Li Jin. Co-phylog: an assembly-free phylogenomic approach for closely related organisms. *Nucleic Acids Research*, 41:e75, 2013.

[134] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18:821–829, 2008.

[135] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M. Karlowski. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*, 18(1):186, 2017.

[136] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

[137] Emile Zuckerkandl and Linus Pauling. Molecules as documents of evolutionary history. *Journal of Theoretical Biology*, 8(2):357 – 366, 1965.

[138] Guanghong Zuo and Bailin Hao. Cvtree3 web server for whole-genome-based and alignment-free prokaryotic phylogeny and taxonomy. *Genomics, Proteomics & Bioinformatics*, 13(5):321 – 331, 2015.

[139] Guanghong Zuo, Zhao Xu, and Bailin Hao. Phylogeny and taxonomy of archaea: A comparison of the whole-genome-based cvtree approach with 16s rrna sequence analysis. *Life*, 5(1):949–968, 2015.

Curriculum Vitae

Name	Chris-Andre Leimeister
Date of birth	February, 10^{th}, 1986 (Ludwigsburg)
Nationality	German
E-Mail	mail@chris-leimeister.de

Education

Since 2015	Göttingen Graduate School for Microbiology and Biochemistry (GGNB), thesis project 'Filtered spaced-word matches: a novel approach to fast and accurate sequence comparison'
2013-2015	University of Göttingen, Master Applied Computer Science with focus on Bioinformatics, thesis 'A Novel Pseudo-Alignment Approach to Fast Genomic Sequence Comparison'
2009-2013	University of Göttingen, Bachelor Applied Computer Science with focus on Bioinformatics, thesis '*kmacs:* the k-Mismatch Average Common Substring Approach for Phylogeny Reconstruction'
2008-2009	University of Göttingen, Business Informatics
2006-2007	University of Augsburg, Economics

Employment history

Since 2015	University of Göttingen (Department of Bioinformatics), scientific assistant
2013-2015	University of Göttingen (Department of Bioinformatics), student worker: development of new alignment-free methods to fast sequence comparison
2013-2014	Max Planck Institute for Dynamics and Self-Organization Göttingen, student programmer: database programming
2007-2008	Karl-Storz GmbH & Co KG., Tuttlingen/Tallin(Estonia), internship
2005-2006	Klinikum des Landkreis Tuttlingen, civilian service.