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

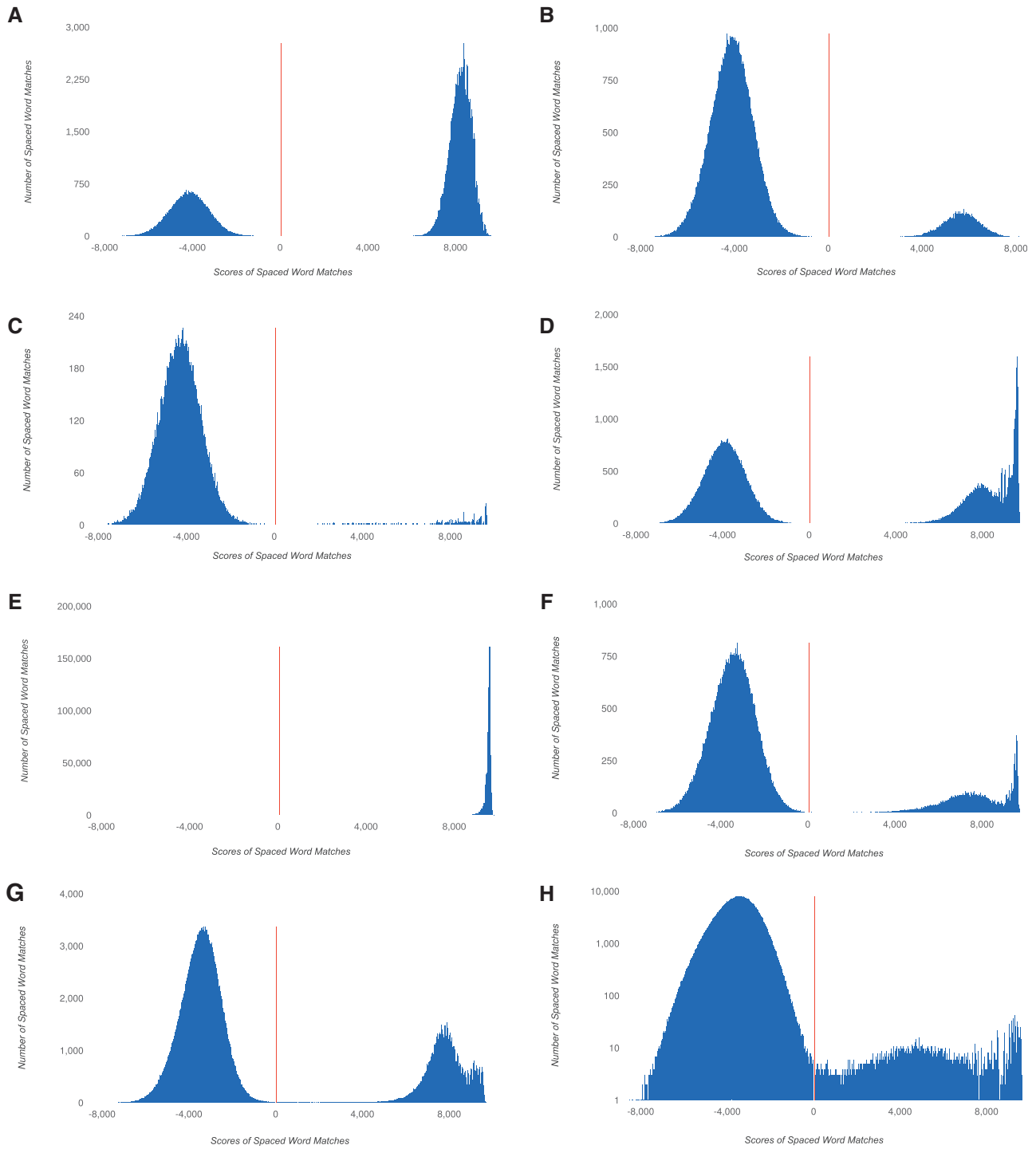

Fig. 1. *Spaced-word histograms* for simulated and real-world sequence pairs. The number of spaced-word matches is plotted against the spaced-word score as defined in the main text. The plots show the remaining spaced-word matches *after* the greedy one-to-one mapping explained in the main text. Thus, 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,

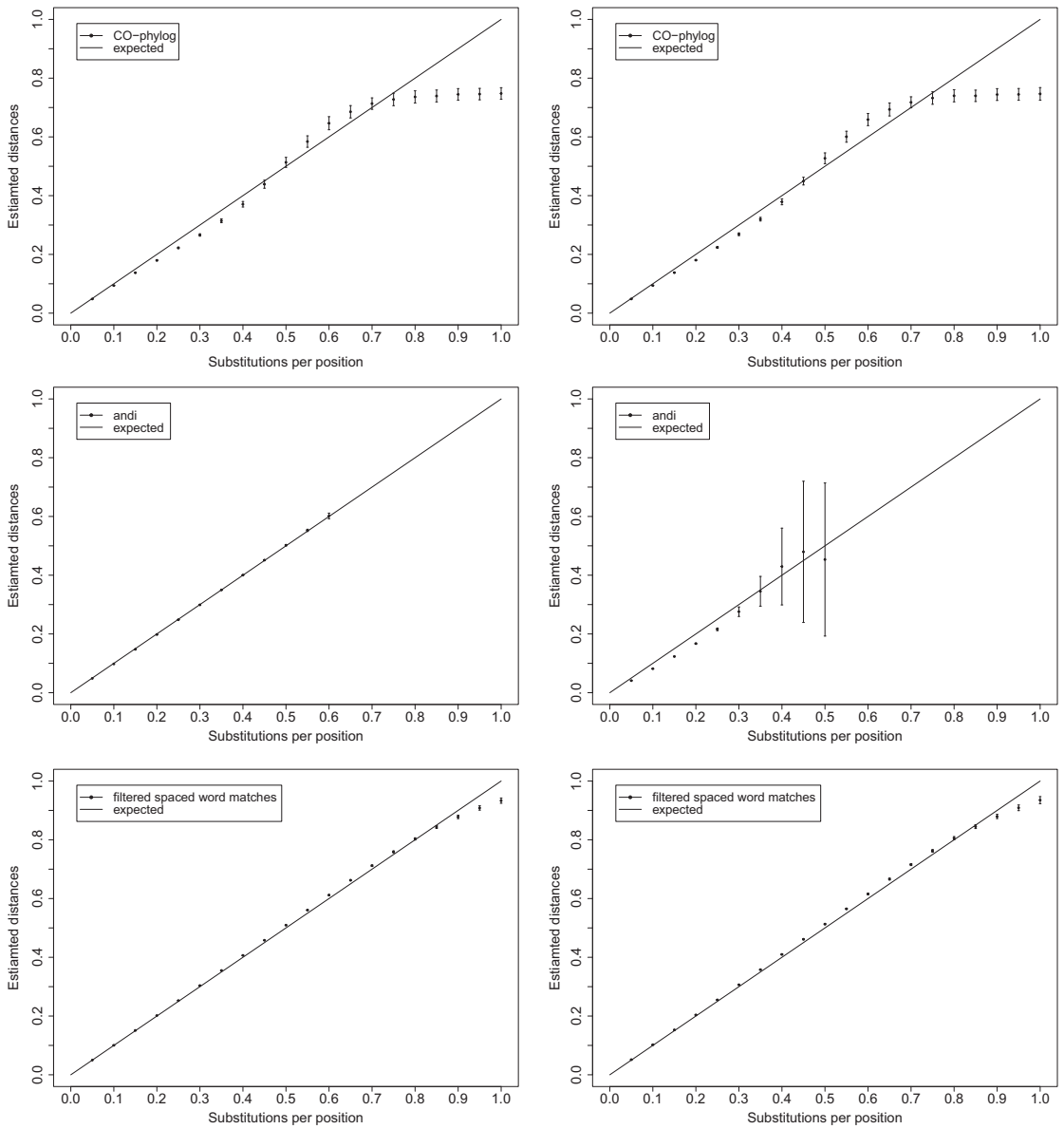

Fig. 2. Distances estimated with *Co-phylog* (top), *andi* (middle) and *FSWM* (bottom) for pairs of simulated DNA sequences, without indels (left-hand side) and with indels (right-hand side), plotted against the 'real' distances, measured in substitutions per site. Sequence pairs were generated as explained 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.

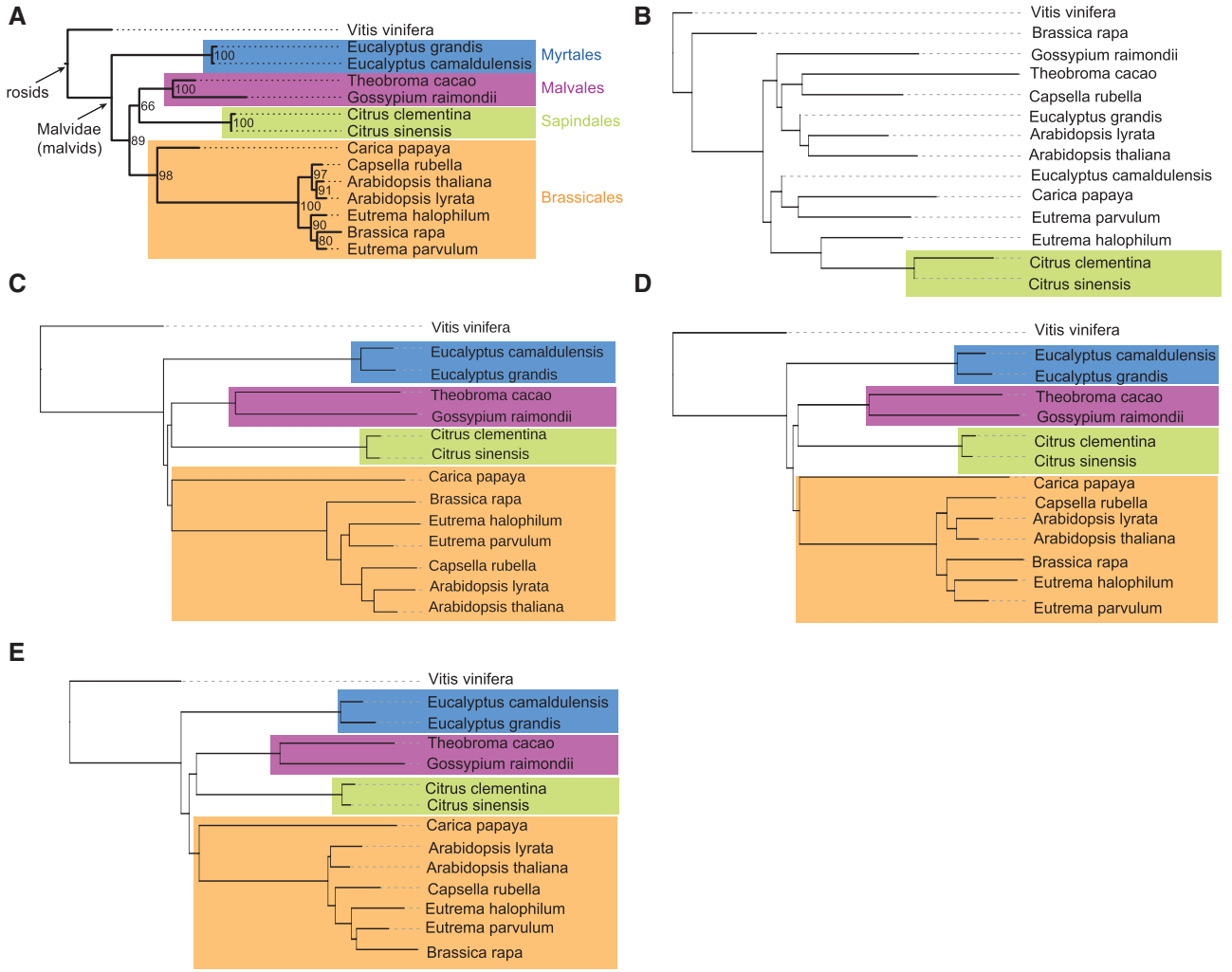

Fig. 3. Trees reconstructed from 14 plant genomes. (**A**) tree based on multiple protein alignments and *Maximum Likelihood* (Hatje and Kollmar, 2012); (**B**) tree calculated with distances from *andi*; (**C–E**) trees calculated with distances from *FSWM* with weights $w = 12$ (C), $w = 13$ (D) and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