Insertions and deletions as phylogenetic signal in an alignment-free context

Niklas Birth[1], Thomas Dencker[1] and Burkhard Morgenstern[1,2,3]

[1]Department of Bioinformatics, Institute of Microbiology and Genetics, Universe'sät Göttingen, Goldschmidtstr. 1
[2]Göttingen Center of Molecular Biosciences (GZMB), Justus-von-Liebig-Weg 11
[3]Campus-Institute Data Science (CIDAS), Goldschmidtstr. 1, 37077 Göttingen, Germany

September 19, 2021

Abstract

Most methods for phylogenetic tree reconstruction are based on sequence alignments; they infer phylogenies from substitutions that may have occurred at the aligned sequence positions. Gaps in alignments are usually not employed as phylogenetic signal. In this paper, we explore an alignment-free approach that uses insertions and deletions (indels) as an additional source of information for phylogeny inference. For a set of four or more input sequences, we generate so-called *quartet blocks* of four putative homologous segments each. For *pairs* of such quartet blocks involving the same four sequences, we compare the distances between the two blocks in these sequences, to obtain hints about indels that may have happened between the blocks since the respective four sequences have evolved from their last common ancestor. A prototype implementation is presented to infer phylogenetic trees from these data, using a *quartet-tree* approach or, alternatively, under the *maximum-parsimony* paradigm. This approach should not be regarded as an alternative to established methods, but rather as a complementary source of phylogenetic information. Interestingly, however, our software is able to produce phylogenetic trees from putative indels alone that are comparable to trees obtained with existing alignment-free methods.

1 Introduction

Most phylogenetic studies are based on multiple sequence alignments (MSAs), either of partial or complete genomes or of individual genes or proteins. If MSAs of multiple genes or proteins are used, there are two possibilities to infer a phylogenetic tree: (1) the alignments can be concatenated to form a so-called *superalignment* or *supermatrix*. Tree building methods such as *Maximum-Likelihood* [44, 14], *Bayesian Approaches* [38] or *Maximum-Parsimony* [9, 11, 46] can then be applied to these superalignments. (2) One can calculate a separate tree for each gene or protein family and then use a *supertree approach* [4] to amalgamate these different trees into one final tree, with methods such as *ASTRAL* [53] or *MRP* [35].

Multiple sequence alignments usually contain gaps representing insertions or deletions (*indels*) that are assumed to have happened since the aligned sequences evolved from their last common ancestor. Gaps, however, are usually not used for phylogeny reconstruction. Most of the above tree-reconstruction methods are based on substitution models for nucleotide or amino-acid residues. Here, alignment columns with gaps are either completely ignored, or gaps are treated as 'missing information', for example in the frequently used tool *PAUP** [46]. Some models have been proposed that can include gaps in a *Maximum-Likelihood* setting, such as *TKF91* [48] and TKF92 [49], see also [18, 1, 28]. Unfortunately, these models do not scale well to genomic data. Thus, indels are rarely used as a source of information for the phylogenetic analysis.

In those studies that actually make use of indels, this additional information is usually encoded in some simple manner. The most straightforward way of doing this is to treat the gap character as a fifth character for DNA comparison, or as a 21st character in protein comparison, respectively. This means that the lengths of gaps are not explicitly considered, so a gap of length $\ell > 1$ is considered to represent ℓ independent insertion or deletion events. Some more issues with this approach are discussed in [40]; these authors introduced the 'simple encoding' of indel data as an alternative. For every indel in the multiple sequence alignment, an additional column is appended. This column contains a present/absent encoding for an indel event which is defined as a gap with given start and end positions. If a longer gap is fully contained in a shorter gap in another sequence, it is considered as *missing information*. Such a simple binary encoding is an effective way of using the length of the indels to gain additional information and can be used in some *maximum-parsimony* framework. A disadvantage of these approaches is their relatively long runtime. The above authors also proposed a more complex encoding of gaps [40] which they further refined in a subsequent paper [32].

The commonly used approaches to encode gaps for phylogeny reconstruction are compared in [33].

The 'simple encoding' of gaps has been used in many studies; one recent study obtained additional information on the phylogeny of Neoaves which was hypothesized to have a 'hard polytomy' [20]. Despite such successes, indel information is still largely ignored in phylogeny reconstruction. Oftentimes, it is unclear whether using indels is worth the large overhead and increased runtime. On the hand, it has also been shown that gaps can contain substantial phylogenetic information [8].

All of the above mentioned approaches to use indel information for phylogeny reconstruction require MSAs of the compared sequences. Nowadays, the amount of the available molecular data is rapidly increasing, due to the progress in next-generation sequencing technologies. If the size of the analyzed sequences increases, calculating multiple sequence alignments quickly becomes too time consuming. Thus, in order to provide faster and more convenient methods to phylogenetic reconstruction, many alignment-free approaches have been proposed in recent years. Most of these approaches calculate pairwise distances between sequences, based on sequence features such as k-mer frequencies [41, 34, 22] or the number [26] or length [23, 50, 30] of word matches. Distance methods such as *Neighbor-Joining* [39] or *BIONJ* [12] can then reconstruct phylogenetic trees from the calculated distances. For an overview, the reader is referred to recent reviews of alignment-free methods [51, 16, 3].

Some recently proposed alignment-free methods use inexact word matches between pairs of sequences [52, 17, 25], where mismatches are allowed to some degree. Such word matches can be considered as pairwise, gap-free 'mini-alignments'. So, strictly spoken, these methods are not 'alignment-free'. In the literature, they are still called 'alignment-free', as they circumvent the need to calculate full sequence alignments of the compared sequences. The advantage of such 'mini-alignments' is that inexact word matches can be found almost as efficiently as exact word matches, by adapting standard word-matching algorithms.

A number of these methods use so-called *spaced-words* [19, 22, 31]. A spaced-word is a word that, in addition to nucleotide or amino-acid symbols, contains *wildcard* characters at certain positions that are specified by a pre-defined binary pattern P representing 'match positions' and 'don't-care positions', see Figure 1 for an example. If the same 'spaced word' occurs in two different sequences, this is called a *Spaced-word Match* or *SpaM*, for short. One way of using spaced-word matches – or other types of inexact word matches – in alignment-free sequence comparison is to use them as a proxy for full alignments, to estimate the number of mismatches per position

in the (unknown) full sequence alignment. This idea has been implemented in the software *Filtered Spaced Word Matches (FSWM)* [25]; it has also been applied to protein sequences [24], and to unassembled reads [21]. Other approaches have been proposed recently, that use the *number* of *SpaMs* to estimate phylogenetic distances between DNA sequences [31, 37], see [29] for a review of the various *SpaM*-based methods.

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

Figure 1: Binary pattern $P =$ '110101′ representing *match positions* ('1') and *don't-care positions* ('0') and a *spaced word* '**A G * C * A**' with respect to P, occurring in sequences S_1 and S_2. The occurrence of the same spaced word in two different sequences is called a *Spaced-word Match (SpaM)*.

Multi-SpaM [7] is a recent extension of the *SpaM* approach to *multiple* sequence comparison. For a set of four or more input sequences, and for a binary pattern P, *Multi-SpaM* finds occurrences of the same spaced word with respect to P in *four* different input sequences. Such a spaced-word match is called a *quartet P-block*, or *quartet block*, for short. A *quartet block*, thus, consists of four occurrences of the same spaced-word, with respect to a specific pattern P, as in Figure 2. For each such block, *Multi-SpaM* identifies an optimal quartet tree topology based on the nucleotides aligned to each other at the *don't-care* position of P, using the program *RAxML* [44]. Finally, the quartet trees calculated in this way are used to find a supertree of the full set of input sequences. To this end, *Multi-SpaM* uses the program *Quartet MaxCut* [43].

In the present paper, we use *pairs* of *quartet blocks* involving the same four sequences. We consider the distances between two blocks in the four sequences, to obtain hints about potential insertions and deletions that may have occurred between two quartet blocks. If these distances are different for two of the sequences, this would indicate that an insertion or deletion has happened since these sequences evolved from their last common ancestor. The distances between two quartet blocks can therefore support one of three possible quartet topologies for the four involved sequences. If, for example, in a pair of quartet blocks involving sequences S_i, S_j, S_k, S_l, the distance between these blocks is equal in S_i and S_j as well as in S_k and S_l but the distance in S_i and S_j is different from the one in S_k and S_l, this would support a quartet tree where S_i and S_j are neighbours, as well as S_k and S_l;

an example is shown in Figure 3.

To evaluate the phylogenetic signal that is contained in such pairs of quartet blocks, we first evaluate the inferred quartet topologies directly, by comparing them to trusted reference trees. Next, we use two different methods to infer a phylogenetic tree for the full set of input sequences, based on the distances between quartet blocks. (A) We calculate super trees based on the inferred quartet trees using the software *Quartet MaxCut*. (B) We use distances between pairs of blocks as characters in a *maximum-parsimony* setting, to find a tree that minimizes the number of insertions and deletions that have to be assumed, given the different distances between the quartet blocks. We evaluate these approaches on data sets that are commonly used as benchmark data in alignment-free sequence comparison. Our evaluation shows that the majority of the inferred quartet trees is correct and should therefore be useful additional information for phylogeny reconstruction. Moreover, the quality of the trees that we can infer from our quartet block pairs alone is roughly comparable to the quality of trees obtained with existing alignment-free methods.

The goal of our study is to show that insertions and deletions can be used as phylogenetic signal in an alignment-free context. Note that the information from putative indels is *complementary* to the information used in standard phylogeny approaches where aligned residues are used to infer substitutions that may have happened in the evolution of the sequences. Consequently, our approach is not competing with these existing methods but may be used as *additional* evidence that might support or call into question phylogenies inferred by more traditional approaches.

2 Design and Implementation

2.1 Spaced words, quartet blocks and distances between quartet blocks

We are using standard notation from stringology as defined, for example, in [15]. For a sequence S over some alphabet, $S(i)$ denotes the i-th symbol of S. In order to investigate the information that can be obtained from putative indels in an alignment-free context, we use the *P-blocks* generated by the program *Multi-SpaM* [7]. At the start of every run, a binary pattern $P \in \{0,1\}^{\ell}$ is specified for some integer ℓ. Here, a "1" in P denotes a *match position*, a "0" stands for a *don't-care position*. The number of *match positions* in P is called its *weight* and is denoted by w. By default, we are using parameter values $\ell = 110$ and $w = 10$, so by default the pattern P has

100 *don't-care* positions.

A *spaced word* W with respect to a pattern P is a word over the alphabet $\{A, C, G, T\} \cup \{*\}$ of the same length as P, and with $W(i) = *$ if and only if i is a *don't care position* of P, i.e. if $P(i) = 0$. If S is a sequence of length N over the nucleotide alphabet $\{A, C, G, T\}$, and W is a spaced word, we say that W *occurs* at some position $i \in \{1, \ldots, \ell\}$, if $S(i + j - 1) = W(j)$ for every match position j in P. For two sequences S and S' and positions i and i' in S and S', respectively, we say that there is a *spaced-word match (SpaM)* between S and S' at (i, i'), if the same spaced word W occurs at i in S and at i' in S'. A *SpaM* can be considered as a local pairwise alignment without gaps. Given a nucleotide substitution matrix, the *score* of a spaced-word match is defined as the sum of the substitution scores of the nucleotides aligned to each other at the *don't-care* positions of the underlying pattern P. In *FSWM* and *Multi-SpaM*, we are using a substitution matrix described in [6]. In *FSWM*, only *SpaMs* with positive scores are used. It has been shown that this *SpaM-filtering* step can effectively eliminate most random spaced-word matches [25].

For a set of ≥ 4 input sequences and a binary pattern P of length ℓ, the program *Multi-SpaM* is based on *quartet (P)-blocks*, where a quartet block is defined as four occurrences of some spaced word W in four different sequences, see Figure 2 for an example. A quartet block B can, thus, be considered as a local gap-free four-way alignment, aligning length-ℓ segments of four sequences; we say that B 'involves' these four sequences. To exclude spurious random quartet blocks, *Multi-SpaM* removes quartet blocks with a low degree of similarity between the aligned segments. Technically, a quartet block is required to contain one occurrence of the spaced-word W, such that the other three occurrences of W have positive similarity scores with this first occurrence. For a given nucleotide substitution matrix, the similarity score of two spaced words (with respect to the same pattern P) is defined as the sum of the substitution scores of the nucleotides aligned to each other at the *don't-care* positions of P.

In this paper, we are considering *pairs* of quartet blocks involving the same four sequences, and we are using the distances between the two blocks in these sequences as phylogenetic signal.

2.2 Phylogeny inference using distances between quartet blocks

Let us consider two *quartet blocks* B_1 and B_2 with respect to patterns P_1 and P_2, respectively, involving the same four sequences S_i, S_j, S_k, S_l. We assume

that B_1 is strictly to the left of B_2, in the sense that the last position of B_1 is smaller than the first position of B_2 in all four sequences. Next, let D_ι be the distance between B_1 and B_2 in sequence $S_\iota, \iota = i, \ldots, l$. More formally, if in sequence S_ι block B_1 starts at position k_1 and block B_2 starts at position k_2, then we define D_ι to be $k_2 - k_1 - \ell_1$, where ℓ_1 is the length of the pattern P_1. In other words, D_ι is the length of the segment between B_1 and B_2 in S_ι, see Figures 3 and 4 for examples. We assume that the blocks B_1 and B_2 are representing true homologies, i.e. for each of them the respective segments go back to a common ancestor in evolution. Then, if we find for two sequences, say S_i and S_j, that their distances D_i and D_j between B_1 and B_2 are different from each other, this would imply that at least one insertion or deletion must have happened since S_i and S_j have evolved from their last common ancestor. If, by contrast, the $D_i = D_j$ holds, no such insertion or deletion needs to be assumed.

There are three possible fully resolved (i.e. binary) quartet topologies for the four sequences $S_i, \ldots, S_l$ that we denote by $S_iS_j|S_kS_l$ etc. In the sense of the *parsimony* paradigm, we can consider the distance between two blocks as a *character* and D_ι as the corresponding *character state* associated with sequence S_ι. If two distances, say D_i and D_j, are equal, and the other two distances, D_k and D_l are also equal to each other, but different from S_i and S_j, respectively, this would support the tree topology $S_iS_j|S_kS_l$: with this topology, one would have to assume only one insertion or deletion to explain the character states, while for $S_iS_k|S_jS_l$ or $S_iS_l|S_jS_k$, two insertions or deletions would have to be assumed. In this situation – i.e. if we have $D_i = D_j \neq D_k = D_l$ –, we say that the pair (B_1, B_2) *strongly* supports topology $S_iS_j|S_kS_l$.

Next, we consider the situation where two of the distances are equal, say $D_i = D_j$, and D_k and D_l would be different from each other, and also different from D_i and D_j. From a parsimony point-of-view, all three topologies would be equally good in this case, since each of them would require two insertions or deletions. It may still seem more plausible, however, to prefer the topology $S_iS_j|S_kS_l$ over the two alternative topologies. In fact, if we would use a simple probabilistic model where an insertion/deletion event has a fixed probability p, with $0 < p < 0.5$, along each branch of the topology, then it is easy to see that the topology $S_iS_j|S_kS_l$ would have a higher likelihood than the two alternative topologies. In this situation, we say that the pair (B_1, B_2) *weakly* supports the topology $S_iS_j|S_kS_l$. Finally, we call a pair of quartet blocks *informative*, if it – strongly or weakly – supports one of the three quartet topologies for the involved four sequences.

For a set of input sequences $S_1, \ldots, S_N, N \geq 4$, we implemented two different ways of inferring phylogenetic trees from quartet-block pairs. With

Sequence													
S_1	T	A	G	A	T	G	C	C	A	T	A	A	T
S_2	T	T	A	C	**A**	**G**	G	**C**	A	**A**	T	C	
S_3	G	C	**A**	**G**	A	**C**	G	**A**	C	G	C		
S_4	T	A	G	A	C	A	A	G	T	C	C	T	
S_5	A	T	T	**A**	**G**	G	**C**	C	**A**	C	T	C	A
S_6	A	A	G	G	C	A	A	C	T	C	G		
S_7	A	**A**	**G**	T	**C**	A	**A**	C	T	C	G	T	
S_8	G	C	T	A	G	C	T	T	C	A	T	C	A
S_9	C	T	T	A	A	A	C	G	G	C	T	T	

Figure 2: *Quartet block* with respect to the binary pattern 110101 representing *match positions* ('1') and *don't-care* positions ('0'). The shown *quartet block* involves sequences S_2, S_3, S_5, S_7; the spaced word '**A G * C * A**' occurs in all four sequences. A *quartet block* can be seen as a local, gap-free four-way alignment with matching residues at the *match positions* and possible mismatches at the *don't-care positions* of the underlying binary pattern. Note that this is a toy example, in practice we are using binary patterns of length 110 with 10 *match* and 100 *don't-care* positions.

the first method, we calculate the *quartet topology* for each quartet-block pair that supports one of the three possible quartet topologies. We then calculate a *supertree* from these topologies. Here, we use the program *Quartet MaxCut* [42, 43] that we already used in our previous software *Multi-SpaM* where we inferred quartet topologies from the nucleotides aligned at the *don't-care* positions of quartet blocks.

Our second method uses the distances between quartet blocks as input for *Maximum-Parsimony* [9, 11]. To this end, we generate a character matrix as follows: the rows of the matrix correspond, as usual, to the input sequences, and each informative quartet block pair corresponds to one column. The distances between the two quartet blocks are encoded by characters '0', '1' and '2', such that equal distances in an informative quartet-block pair are encoded by the same character (this encoding is necessary, since some parsimony programs accept only simple characters as input, so we cannot use the distances themselves as characters in the matrix). For sequences not involved in a quartet-block pair, the corresponding entry in the matrix is empty and is considered as 'missing information'. In Figure 3, for example, the entries for S_2, S_4, S_5, S_8 would be '0', '1', '0', '1', respectively; in Figure 4, the entries for S_1, S_4, S_5, S_6 would be '0', '0', '1', '2'.

Figure 5 shows an informative block pair, a character matrix encoding the distances D_i for this block pair in the first column and the distances for three additional hypothetical block pairs in columns 2 to 4, together with

Sequence																	Distance D_i
S_1	A	A	G	A	C	G	T	T	A	C	A	C	G	A	T		
S_2	T	C	**A**	**G**	G	**C**	A	A	**C**	G	**G**	**T**	**A**	C			$D_2 = 2$
S_3	T	T	G	A	G	A	C	A	T	C	C	G	A	T	C	A	
S_4	C	**A**	**G**	A	**C**	A	C	T	**C**	G	**G**	**T**	**A**	T	A		$D_4 = 3$
S_5	A	A	T	A	**A**	**G**	T	**C**	A	T	**C**	A	**G**	**T**	**A**		$D_5 = 2$
S_6	G	A	C	T	C	G	T	T	C	C	C	G	A	C	A		
S_7	G	T	G	C	C	A	A	C	C	C	A	G	C	C	C		
S_8	C	G	**A**	**G**	T	**C**	A	A	T	**C**	A	**G**	**T**	**A**	C	T	$D_8 = 3$
S_9	A	C	C	C	T	C	C	C	G	A	G	C	A	C	A	A	

Figure 3: Two quartet blocks B_1 and B_2 (in green and purple) with respect to binary patterns 1101 and 10111, and with the matching spaced words '**A G * C**' and '**C * G T A**', respectively, involving the same four sequences S_2, S_4, S_5, S_8. The distances between B_1 and B_2 in these sequences are $D_2 = D_5 = 2$ and $D_4 = D_8 = 3$. In the sense of *maximum parsimony*, these distances would support the quartet topology $S_2S_5|S_4S_8$, since this topology would require only one insertion/deletion (indel) event to explain the distances D_i while the alternative two quartet topologies for the involved sequences would require two indel events. With our terminology, we say that this topology is *strongly* supported by the four distance values D_i.

a tree topology inferred from this matrix with *maximum parsimony*. Here, we used the the program *pars* form the *PHYLIP* package [10]. Note that all four block pairs in the matrix *strongly* support one of the three possible quartet topologies, since a block pair that only weakly supports a topology would not be informative in the sense of the parsimony principle. Therefore, in each of the four block pairs, we have only two different distances, and we need only two characters, '0' and '1'.

In order to find suitable quartet-block pairs for the two described approaches, we are using our software *Multi-SpaM*. This program samples up to 1 million quartet blocks. We use the quartet blocks generated by *Multi-SpaM* as *reference blocks*, and for each reference block B_1, we search for a second block in a window of L nucleotides to the right of B_1 for a second block B_2 involving the same four sequences. We use the first block that we find in this window, provided that the involved spaced-word matches are *unique* within the window. If the pair (B_1, B_2) supports a topology of the involved four sequences – either strongly or weakly –, we use this block pair, otherwise the pair (B_1, B_2) is discarded.

Sequence															Distance D_i
S_1	G	**A**	**G**	G	**C**	A	A	**C**	G	**G**	**T**	**A**	C	T	T	$D_1 = 2$
S_2	G	G	A	C	A	C	G	T	T	A	C	C	G	A		
S_3	T	T	G	A	G	A	C	A	T	C	C	G	A	T	C	
S_4	A	A	T	A	**A**	**G**	T	**C**	A	T	**C**	A	**G**	**T**	**A**	$D_4 = 2$
S_5	C	**A**	**G**	A	**C**	A	A	C	T	**C**	G	**G**	**T**	**A**		$D_5 = 4$
S_6	C	G	**A**	**G**	T	**C**	A	A	T	**C**	A	**G**	**T**	**A**	C	$D_6 = 3$
S_7	G	A	C	T	C	G	T	T	C	C	C	G	A	C	A	
S_8	C	T	C	G	T	T	C	C	C	G	A	C	A	A		

Figure 4: Two quartet blocks, similar as in Figure 3, but involving S_1, S_4, S_5, S_6, and with distances $D_1 = D_4 = 2, D_5 = 3$ and $S_6 = 4$. Here, we say that the distances *weakly* support the topology $S_1S_4|S_5S_6$, since only D_1 and D_4 are equal, while D_5 and D_6 are different from each other and from D_1 and D_4.

3 Test results

In order to evaluate the above described approaches to phylogeny reconstruction, we used sets of genome sequences from *AF-Project* [54] that are frequently used as benchmark data for alignment-free methods. In addition, we used a set of 19 *Wolbachia* genomes [13]. For these sets of genomes, trusted phylogenetic trees are available that can be used as reference trees; these genomes have also been used as benchmark data to evaluate *Multi-SpaM* [7].

Note that our indel-based approach is not meant to be an alternative to existing phylogeny approaches that are based on substitutions. Since we are using a complementary source of information, we are not competing with those existing methods, but we wanted to know if our approach might be useful as an *additional* input for tree reconstruction. The comparison with alternative alignment-free phylogeny methods in this section is not done to find out which approach performs better – we rather wanted to find out if or to what extent our indel-based approach can provide relevant information for phylogeny inference at all.

3.1 Quartet trees from quartet-block distances

First, we tested, how many *informative* quartet block pairs we could find, i.e. how many of the identified quartet-block pairs would either *strongly* or *weakly* support one of the three possible quartet topologies for the corresponding four sequences.

For each set of genome sequences, we first sampled 1,000,000 quartet blocks with *Multi-SpaM* [7], we call these blocks the 'reference blocks'. For

(A)

Sequences														Distances D_i
S_1	T	C	A	G	G	C	A	C	G	G	T	A	C	$D_1 = 2$
S_2	A	T	A	A	G	T	A	T	C	A	G	T		$D_2 = 2$
S_3	C	A	G	A	C	A	T	C	G	G	T	A	T	$D_3 = 3$
S_4	T	T	G	A	G	A	C	A	T	C	G	A	T	
S_5	G	A	C	T	C	T	T	C	C	C	G	A		
S_6	C	A	G	A	G	T	A	A	T	C	A	G	T	$D_6 = 3$

(B)

Sequences	Encoded distances D_i between four block pairs			
S_1	1	-	1	1
S_2	1	-	-	0
S_3	0	0	-	-
S_4	-	1	0	1
S_5	-	1	0	0
S_6	0	0	1	-

(C)

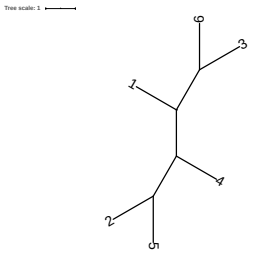

Figure 5: **(A)** Single block pair in a set of 6 sequences and distances D_i, **(B)** character matrix encoding distances D_i from four different quartet-block pairs and **(C)** tree topology, calculated from this matrix with *maximum parsimony*. Each column in the matrix represents one informative block pair. For the four sequences involved in a block pair, the distances D_i are represented by characters '0' and '1', such that equal distances are represented by the same character. The characters themselves are arbitrary, the matrix only encodes if the distances D_i between two blocks are equal or different in the four involved sequences. Dashes in a column represent 'missing information', for sequences that are not involved in the respective quartet-block pair. The quartet-block pair in **(A)** would be represented by the first column of the matrix **(B)**, as we have $D_1 = D_2 \neq D_3 = D_6$. Thus, for S_1 and S_2 we have the same (arbitrary) symbol '1', while S_3 and S_6 we have the symbol '0'. Since S_4 and S_5 are not involved in this quartet-block pair, they have dashes in the first column, representing 'missing information'. The matrix represents four quartet-block pairs that *strongly* support one quartet topology, namely column 1 supporting $S_1S_2|S_3S_6$, column 2 supporting $S_3S_6|S_4S_5$, column 3 supporting $S_1S_6|S_4S_5$ and column 4 supporting $S_1S_4|S_2S_5$.

each of these blocks, we then searched for a second block in a window of 500 *nt* to the right of the reference block. For the second block, we used a pattern $P = 1111111$, i.e. we generated blocks of exact word matches of length seven. If no second block could be found in the window, the reference block was discarded. Table 1 shows the percentage of informative quartet block pairs, among the quartet block pairs that we used. To evaluate the correctness of the obtained quartet topologies, we compared them to the topologies of the respective quartet sub-trees of the reference trees using the *Robinson-Foulds (RF)* distance [36] between the two quartet topologies. If the *RF* distance is zero, the inferred quartet topology is in accordance with the reference tree. To compare the obtained quartet topologies to the full reference trees, we used Sarah Lutteropp's program *Quartet Check* that is available through GitHub [27]. We slightly modified the original code to adapt it to our purposes; the modified code used in our study is also available through GitHub [5].

We want to use the quartet trees that we obtain from informative quartet block pairs, to generate a tree of the full set of input sequences. Therefore, it is not sufficient for us to have a high percentage of correct quartet trees, but we also want to know how many of the sequence quartets are covered by these quartet trees. Generally, the results of super-tree methods depend on the *coverage* of the used quartet topologies [2, 45]. For a set of N input sequences, there are $\binom{N}{4}$ possible 'sequence quartets', i.e. sets of four sequences. Ideally, for every such set, we should have at least one quartet tree, in order to find the correct super tree. Table 1 reports the *quartet coverage*, i.e. the percentage of all sequence quartets, for which we obtained at least one quartet tree.

Note that *Multi-SpaM* uses randomly sampled quartet blocks, the program can thus return different results for the same set of input sequences. We therefore performed 10 program runs on each set of sequences and report the average *correctness* and *coverage* of these test runs.

3.2 Full phylogeny reconstruction

Finally, we applied our quartet-block pairs to reconstruct full tree topologies for the above sets of benchmark sequences. Here, we used two different approaches, namely *Quartet MaxCut* and *Maximum-Parsimony*, as described above. As is common practice in the field, we evaluated the quality of the reconstructed phylogenies by comparing the the respective reference trees from *AFproject* using the *normalized Robinson-Foulds (RF) distances* between the

	Strong support			Weak support			Strong & weak combined		
	# info	% corr	% cov	# info	% corr	% cov	# info	% corr	% cov
E. coli 27	54,713	78.57	72.68	17,613	54.40	47.18	72,326	72.69	79.7
E. coli 29	54,752	80.46	64.27	15,844	56.49	36.89	70,596	75.08	72.48
Fish mito	7,425	66.98	25.93	10,497	59.10	35.17	17,922	62.37	46.58
Yersinia	5,675	43.84	100.00	5,723	41.53	100.00	11,398	42.68	100.00
Plants	15,617	82.67	84.05	99,178	76.70	99.93	114,795	77.51	99.96
Wolbachia	95,936	92.52	94.62	37,633	72.03	87.09	133,569	86.75	98.97

Table 1: Test results on different sets of genomes. As benchmark data, we used four sets of genome sequences from *AF-Project* [54] and one set of *Wolbachia* genomes [13]. For each data set, we generated up to 1,000,000 pairs of quartet blocks as described in the main text. The table shows the number of *informative* block pairs ('# info'), i.e. the number of block pairs for which we obtained either strong or weak support for one of the three possible quartet topologies of the involved sequences. In addition we show the percentage of *correct* quartet topologies (with respect to the respective reference tree), out of all informative block pairs, as well as the 'coverage' by quartet blocks, i.e. the percentage of sequence quartets for which we found at least one informative block pair. For each data set, we obtained 1,000,000 block pairs, except for the 'Fish Mito' data set, where our method could find only 42,765 block pairs, on average.

inferred and the reference topologies. For a data set with N taxa, the *normalized RF distances* are obtained from the *RF distances* by dividing them by $2 * N - 6$, i.e. by the maximum possible *distance* for trees with n leaves. The results of other alignment-free methods on these data are reported in [7, 54].

We applied the program *Quartet MaxCut* first to the quartet topologies derived from the set of *all* informative quartet-block pairs. As a comparison, we then inferred topologies using only those quartet-block pairs that *strongly* support one of the three possible topologies for the four involved sequences. The results of these test runs are shown in Table 2.

Next, we used the program *PAUP** [46] to calculate the most parsimonious tree, using the distances between quartet blocks as characters, as explained above. Here, we used the *TBR* [47] heuristic. In some cases, this resulted in multiple optimal, i.e. most parsimonious trees. In these cases, we somewhat arbitrarily picked the first of these trees in the *PAUP** output. The results of these test runs are also shown in Table 2, together with the results from *Multi-SpaM*.

		Indel Information					
		Quartet MaxCut			Parsimony	*Multi-SpaM*	*FSWM*
	# seq	Strong	Weak	Combined			
E. coli 29	29	12.6	22.3	12.6	11.2	12.6	6
E. coli 27	27	9.8	18.7	10.8	6.2	8.8	8
Fish mito	25	20.6	26	19.2	21.2	7.8	2
Yersinia	8	6	8.6	6	6	6.2	10
Plants	14	6.8	7.6	6.8	6.2	6	6
Wolbachia	19	5.8	8	6.8	5.8	6	6

Table 2: Average *Robinson-Foulds (RF)* distances between trees, reconstructed with various alignment-free methods, for five sets of genome sequences from *AFproject* [54] and one set of genomes from *Wolbachia*. For each data sets, the average over 10 program runs was taken.

4 Discussion

Sequence-based phylogeny reconstruction usually relies on nucleotide or amino-acid residues aligned to each other in multiple alignments. Information about insertions and deletions (indels) is neglected in most studies, despite evidence that this information may be useful for phylogeny inference. There are several difficulties when indels are to be used as phylogenetic signal: it is difficult to derive probabilistic models for insertions and deletions, and there are computational issues if gaps of different lengths are spanning multiple columns in multiple alignments. Finally, gapped regions in sequence alignments are often considered less reliable than un-gapped regions, so the precise number and length of insertions and deletions that have happened may not be easy to infer from multiple alignments.

In recent years, many fast alignment-free methods have been proposed to tackle the ever increasing amount of sequence data. Most of these methods are based on counting or comparing *words*, and gaps are usually not allowed within these words. It is therefore not straight-forward to adapt standard alignment-free methods to use indels as phylogenetic information.

In the present paper, we proposed to use *pairs of blocks* of aligned sequence segments to obtain information about possible insertions and deletions since the compared sequences have evolved from a common ancestor. *Within* such blocks, no gaps are allowed. To obtain hints about possible insertions or deletions, we consider the distances between these blocks in the respective sequences. If these distances are different for two sequences, this indicates that there has been an insertion or deletion since they evolved from their last common ancestor. If, by contrast, the two distances are the same, no

indel event needs to be assumed. This information can be used to infer a tree topology for the sequences involved in a pair of blocks. To our knowledge, this is the first attempt to use insertions and deletions as phylogenetic signal in an alignment-free context.

In this study, we restricted ourselves, for simplicity, to *quartet blocks* i.e. to blocks involving *four* input sequences each, and we used pairs of blocks involving the same four sequences. We did not consider the *length* of hypothetical insertions and deletions, but only asked whether or not such an event has to be assumed between two sequences in the region bounded by the two blocks. Since indels are relatively rare events, compared to substitutions, the *maximum parsimony* paradigm seems to be suitable in this situation. In the sense of *parsimony*, however, only those block pairs are informative that *strongly* support one of three possible quartet topologies, in the sense of the definition that we introduced in this paper. Indeed, if two distances between two blocks are equal, and the third and fourth distance are different from them – and also different from each other –, then each of the three possible quartet topologies would require two insertion or deletion events. That is, all three topologies would be equally good from a parsimonious viewpoint.

Intuitively, however, one may want to use the information from such quartet blocks pairs that, in our terminology, *weakly* support one of the possible topologies. It is easy to see that, with a simple probabilistic model under which an insertion between two blocks occurs with a probability $p < 0.5$, independently of the length of the insertion and the distance between the blocks, a *weakly* supported topology would have a higher likelihood than the two alternative topologies – even if all topologies look equally good from a parsimony point-of-view. So it might be interesting to apply such a simple probabilistic model to our approach, instead of maximum parsimony. Also, while we restricted ourselves to quartet blocks in this study, it might be worthwhile to use block pairs involving more than four sequences.

Our approach has only few parameters that can be adjusted by the user, essentially concerning the underlying binary pattern P and the threshold that is used to separate random quartet blocks form quartet blocks that represent true homologies. In our study, we used patterns with a length of 110 and with 10 match positions, i.e. with 100 don't-care positions. Our results in the present and in previous studies indicate that with our default parameter value, random spaced-word matches can be reliably distinguished from background matches [25, 7]. Adapting these parameter values mainly affects the *number* of quartet-block pairs. So this mainly comes down to a trade off between program run time and the amount of information that one obtains form the block pairs, i.e. the strength of the signal.

Using standard benchmark data, we could show that phylogenetic signal

from putative insertions and deletions between quartet blocks is mostly in accordance with the reference phylogenies that we used as standard of truth. Interestingly, the quality of the tree topologies that we constructed from our 'informative' pairs of quartet blocks – i.e. from indel information alone – is roughly comparable to the quality of topologies obtained with existing alignment-free methods.

As mentioned above, our approach is not competing with existing phylogeny approaches. In fact, we did not expect to obtain trees with our approach that are of comparable quality as trees obtained with standard methods. Our goal was to find out if information about putative insertions and deletions can provide useful phylogenetic information at all in an alignment-free setting. Since the phylogenetic signal from indels is complementary to the information that is used by those existing approaches, any such information might be useful as additional evidence, no matter if substitution-based or indel-based trees are superior. It is all the more surprising that our rather simplistic approach is already able to infer trees that are roughly comparable to trees obtained with established alignment-free approaches.

References

[1] Alexander V. Alekseyenko, Christopher J. Lee, and Marc A. Suchard. Wagner and Dollo: a stochastic duet by composing two parsimonious solos. *Systematic Biology*, 57:772–784, 2008.

[2] Eliran Avni, Zahi Yona, Reuven Cohen, and Sagi Snir. The Performance of Two Supertree Schemes Compared Using Synthetic and Real Data Quartet Input. *J. Mol. Evol.*, 86:150–165, 2018.

[3] 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*, 22:426–435, 2019.

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

[5] Niklas Birth. Single Quartet Check. `https://github.com/njbirth/single_quartet_check`, 2021.

[6] Francesca Chiaromonte, Von Bing Yap, and Webb Miller. Scoring pairwise genomic sequence alignments. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Pacific Symposium on Biocomputing*, pages 115–126, Lihue, Hawaii, 2002.

[7] 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. *NAR Genomics and Bioinformatics*, 2:lqz013, 2020.

[8] Christoph Dessimoz and Manuel Gil. Phylogenetic assessment of alignments reveals neglected tree signal in gaps. *Genome Biol.*, 11:R37, 2010.

[9] James S. Farris. Methods for computing Wagner trees. *Systematic Biology*, 19:83–92, 1970.

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

[11] Walter Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Zoology*, 20:406–416, 1971.

[12] Olivier Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Molecular Biology and Evolution*, 14:685–695, 1997.

[13] 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, 2017.

[14] Stéphane Guindon, Jean-François Dufayard, Vincent Lefort, Maria Anisimova, Wim Hordijk, and Oliver Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.*, 59:307–321, 2010.

[15] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK, 1997.

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

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

[18] Ian H. Holmes. Solving the master equation for Indels. *BMC Bioinformatics*, 18:255, 2017.

[19] 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.

[20] Peter Houde, Edward L. Braun, Nitish Narula, Uriel Minjares, and Siavash Mirarab. Phylogenetic signal of indels and the neoavian radiation. *Diversity*, 11, 2019.

[21] Anna Katharina Lau, Svenja Dörrer, Chris-André Leimeister, Christoph Bleidorn, and Burkhard Morgenstern. *Read-SpaM*: assembly-free and alignment-free comparison of bacterial genomes with low sequencing coverage. *BMC Bioinformatics*, 20:638, 2019.

[22] 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.

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

[24] Chris-Andre Leimeister, Jendrik Schellhorn, Svenja Dörrer, Michael Gerth, Christoph Bleidorn, and Burkhard Morgenstern. Prot-SpaM: Fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *GigaScience*, 8:giy148, 2019.

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

[26] 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*, 99:13980–13989, 2002.

[27] Sarah Lutteropp. Quartet Check. `https://github.com/lutteropp/quartet_check`, 2021.

[28] István Miklós, Gerton A. Lunter, and Ian Holmes. A "Long Indel" Model For Evolutionary Sequence Alignment. *Molecular Biology and Evolution*, 21:529–540, 2004.

[29] Burkhard Morgenstern. Sequence comparison without alignment: The SpaM approaches. In Kazutaka Katoh, editor, *Multiple Sequence Alignment*, Methods in Molecular Biology, pages 121–134. Springer, 2020.

[30] 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.

[31] 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.

[32] Kai Müller. Incorporating information from length-mutational events into phylogenetic analysis. *Mol. Phylogenet. Evol.*, 38(3):667–676, Mar 2006.

[33] T. Heath Ogden and Michael S. Rosenberg. How should gaps be treated in parsimony? A comparison of approaches using simulation. *Mol. Phylogenet. Evol.*, 42:817–826, 2007.

[34] 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.

[35] Mark A. Ragan. Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.*, 1:53–58, 1992.

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

[37] Sophie Röhling, Alexander Linne, Jendrik Schellhorn, Morteza Hosseini, Thomas Dencker, and Burkhard Morgenstern. The number of k-mer matches between two DNA sequences as a function of k and applications to estimate phylogenetic distances. *PLOS ONE*, 15:e0228070, 2020.

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

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

[40] Mark P. Simmons and Helga Ochoterena. Gaps as characters in sequence-based phylogenetic analyses. *Syst. Biol.*, 49:369–381, 2000.

[41] 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.

[42] Sagi Snir and Satish Rao. Quartets MaxCut: A divide and conquer quartets algorithm. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 7:704–718, 2010.

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

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

[45] M. Shel Swenson, Rahul Suri, C. Randal Linder, and Tandy Warnow. An experimental study of Quartets MaxCut and other supertree methods. *Algorithms Mol Biol*, 6:7, 2011.

[46] David Swofford. PAUP*. phylogenetic analysis using parsimony (*and other methods). version 4.0b10. *Sinauer Associates, Sunderland, Massachusetts*, 2003.

[47] David L. Swofford and Garry J. Olsen. Phylogeny reconstruction. In D.M. Hillis and C. Moritz, editors, *Molecular Systematics*, pages 407–511. Sinauer Associates, 1990.

[48] Jeffrey L. Thorne, Hirohisa Kishino, and Joseph Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33:114–124, 1991.

[49] Jeffrey L. Thorne, Hirohisa Kishino, and Joseph Felsenstein. Inching toward reality: An improved likelihood model of sequence evolution. *Journal of Molecular Evolution*, 34:3–16, 1992.

[50] 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.

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

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

[53] Chao. Zhang, Maryam Rabiee, Erfan Sayyari, and Siavash Mirarab. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19(Suppl 6):153, 2018.

[54] Andrzej Zielezinski, Hani Z Girgis, Guillaume Bernard, Chris-André Leimeister, Kujin Tang, Thomas Dencker, Anna Katharina Lau, Sophie Röhling, JaeJin Choi, Michael S Waterman, Matteo Comin, Sung-Hou Kim, Susana Vinga, Jonas S Almeida, Cheong Xin Chan, Benjamin

James, Fengzhu Sun, Burkhard Morgenstern, and Wojciech M Karlowski. Benchmarking of alignment-free sequence comparison methods. *Genome Biology*, 20:144, 2019.