Multi-SpaM: a Maximum-Likelihood approach to Phylogeny Reconstruction based on Multiple Spaced-Word Matches

Md. Nazmul Hasan, Arefin Rahman Niloy

1. Introduction

To gain a better understanding of the evolution of genes or species, reconstructing accurate phylogenetic trees is essential. This can be done using standard methods which rely on *sequence alignments*, either of entire genomes or of sets of orthologous genes or proteins. *Character-based* methods such as *Maximum Parsimony* [14, 20] or *Maximum Likelihood* [15] infer trees based on evolutionary substitution events that may have happened since the species evolved from a common ancestor. These methods are generally considered to be accurate, as long as the underlying alignments are of high quality, and as long as suitable substitution models are used. However, for the task of multiple alignment no exact polynomial-time algorithm exists, and even heuristic approaches can be time consuming [46]. Moreover, the most popular heuristic for multiple alignment, the *progressive alignment* [19], has been shown to be relatively unstable: multiple alignments calculated with progressive approaches and trees inferred from these alignments depend on the underlying *guide trees* and even on the order of the input sequences [9]. In addition to these difficulties, exact algorithms for character-based phylogeny approaches are themselves *NP hard* [11, 21].

[⋆] Fully documented templates are available in the elsarticle package on CTAN.

Email addresses: 0419052003 (Md. Nazmul Hasan), nazmulcse25@gmail.com (Md. Nazmul Hasan), 108052108 (Arefin Rahman Niloy), arefinniloy@gmail.com (Arefin Rahman Niloy)

Distance methods, by contrast, infer phylogenies by estimating evolutionary distances for all pairs of input taxa [16]. Here, pairwise alignments are sufficient which can be faster calculated than multiple alignments, but still require runtime proportional to the product of the lengths of the aligned sequences. There is a loss in accuracy, however, compared to character-based approaches, as all of the information about evolutionary events is reduced to a single number for each pair of taxa, and not more than two sequences are considered simultaneously, as opposed to character-based approaches, where all sequences are examined simultaneously. The final trees are obtained by clustering based on distance matrices, most commonly with *Neighbor Joining* [45]. Since both pairwise and multiple sequence alignments are computationally expensive, they are ill-suited for the increasingly large datasets that are available today due to the next generation sequencing techniques.

In recent years, a number of *alignment-free* approaches to genome-based phylogeny reconstruction have been published which are very fast in comparison to alignment-based methods [49, 57, 5, 7, 40, 42]. Another advantage of these new methods is that they circumvent some well-known problems in genome alignment such as genome rearrangements and duplications. Moreover, alignment-free methods can be applied to incomplete sequence sets and even to collections of unassembled reads [44, 50, 56, 12]. A disadvantage of these methods is that they are, in general, considerably less accurate than slower, alignment-based methods.

Some 'alignment-free' approaches compare fixed-length *words* of the input sequences to each other, so – despite being called 'alignment-free' – they are using local pairwise 'mini-alignments'. Recently, methods have been proposed that estimate phylogenetic distances based on the relative frequency of mismatches in such local alignments. An example is *co-phylog* [56] which finds short gap-free alignments of a fixed length, consisting of matching nucleotide pairs only, except for the middle position where a mismatch is allowed. Phylogenetic distances are estimated from the fraction of such alignments for which the middle position is a mismatch. As a generalization of this approach, *andi* [26] uses pairs of maximal exact word matches that have the same distance to each other in both sequences; the frequency of mismatches in the segments between those matches is then used to estimate the number of substitutions per position between two input sequences.

co-phylog and *andi* require a minimum length of the flanking word matches in order to reduce the number of random background matches. Threfore, they tend not to perform well on distantly related sequences where long exact matches are less frequent. Moreover, the number of random segment matches grows quadratically with the length of the input sequences while the expected number of homologous matches grows only linearly. Thus, longer exact matches are necessary in these approaches to limit the number of background matches if longer sequences are compared. This, in turn, reduces the number of homologies that are found, and therefore the amount of information that can be used to calculate accurate distances. Other alignment-free approaches are based on the length of maximal common substrings between sequences. These approaches are also very efficient, since common substrings can be rapidly found using suffix trees or related data structures [55, 27]. As a generalization of this approach, some methods use longest common substrings with a certain number of mismatches [30, 54, 53, 33, 3].

Recently, we proposed to use words with *wildcard characters* – so-called *spaced words* – for alignment-free sequence comparison [29, 28]. Here, a binary pattern of *match* and *don't-care* positions specifies the positions of the *wildcard* characters [38, 36, 23]. In *Filtered Spaced-Word Matches (FSWM)* [32] and *Proteome-based Spaced-word Matches (Prot-SpaM)* [31], alignments of such spaced words are used, where sequence positions must match at the *match* positions while mismatches are allowed at the *don't care positions.* A score is calculated for every such spaced-word match in order to remove – or *filter out* – *background* spaced-word matches; the mismatch frequency of the remaining *homologous* spaced-word matches is then used to estimate the number of substitutions per position that happened since two sequences evolved from their last common ancestor. The filtering step allows us to use patterns with fewer match positions in comparison to above mentioned methods *co-phylog* and *andi*, since the vast majority of the background noise can be eliminated reliably by looking at the *don't-care* positions of the initially found spaced-word matches. As a result, the phylogenetic distances calculated by *FSMW* and *Prot-SpaM* are generally rather accurate, even for large and distantly related sequences.

In this paper, we introduce a novel approach to phylogeny reconstruction called *Multiple Spaced-Word Matches (Multi-SpaM)* that combines the *speed* of the so-called 'alignment-free' methods with the *accuracy* of the *Maximum-Likelihood* approach. While other alignment-free methods are limited to *pairwise* sequence comparison, we generalize the above outlined *spaced-word* approach to *multiple* sequence comparison. For a binary pattern of *match* and *don't care* positions, *Multi-SpaM* identifies *quartet blocks* of four matching spaced words each, *i.e.* gap-free four-way alignments with matching nucleotides at the *match* positions of the underlying binary pattern and possible mismatches at the *don't care* positions, see Figure **??** for an example. For each such quartet block, an optimal *Maximum-Likelihood* tree topology is calculated with the software *RAxML* [51]. The *Quartet MaxCut* algorithm [48] is then used to combine the calculated quartet tree topologies into a super tree. We show that on both simulated and real data, *Multi-SpaM* produces phylogenetic trees of high quality and often outperforms other alignment-free methods.

2. Material and Method

To describe our method, we first need some formal definitions. A *spaced word* of length ℓ exists in the context of a binary pattern $P \in \{0,1\}^\ell$ of the same length. This pattern marks every position as either a *match position* in case of a 1 or as a *don't care position* in case of a 0. The number of match positions is called the *weight* of the pattern. Given such a pattern P, a *spaced word* w is a word of length ℓ over the alphabet {A,C,G,T,*} such that $w(i) = *$ if and only if $P(i) = 0$, *i.e.* if and only if i is a *don't care* position. The symbol '$*$' is interpreted as a 'wildcard' character. For a DNA Sequence S of length n

and a position $0 \leq i \leq n - l + 1$, we say that a *spaced word* w occurs in S at position i – or that $[S, i]$ is an *occurrence* of w – if $S(i + j) = w(j)$ for all match positions j. This follows the definition that we previously used [29, 34].

A pair $([S, i], [S', i'])$ of occurrences of the same spaced word w is called a *spaced-word match.* For a substitution matrix assigning a *scores*(X, Y) to every pair (X, Y) of nucleotides, we define the *score* of a spaced word match $([S, i], [S', i'])$ as

$$\sum_{P(k)=0} s(S(i + k), S'(i' + k))$$

That is, if we align the two occurrences of w to each other, the score of the spaced-word match is the sum of the scores of the nucleotides aligned to each other at the *don't-care* positions of P. In *Multi-SpaM*, we are using the nucleotide substitution matrix below that has been proposed by Chiaromonte *et al.* [10]:

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

Multi-SpaM starts with generating a binary pattern P with user-defined length ℓ and weight w. By default, we use values $\ell = 110$ and $w = 10$, *i.e.* by default the pattern has 10 *match positions* and 100 *don't-care* positions, but other values for ℓ and w can be chosen by the user. Given these parameters, P is calculated by running our previously developed software tool *rasbhari* [24].

As a basis for phylogeny reconstruction, we are using four-way alignments consisting of occurrences of the same spaced word with respect to P in four different sequences. We call such an alignment a *quartet P-block* or a *P-block*, for short. A P-block is thus a gap-free alignment of length ℓ where in the k-th column identical nucleotides are aligned if k is a *match* position in P, while mismatches are possible if k is a *don't-care* position, see Figure **??** for an example. Note that the number of such P-blocks can be very large: if there are

n occurrences of a spaced-word w in n different sequences, then this gives rise to $\binom{n}{4}$ different P-blocks. Thus, instead of using all possible P-blocks, *Multi-SpaM* randomly samples a limited number of P-blocks to keep the program runtime under control.

Moreover, for phylogeny reconstruction, we want to use P-blocks that are likely to represent true homologies. Therefore, we introduce the following definition: a P-block – *i.e.* a set of four occurrences of the same spaced word w – is called a *homologous P-block* if it contains at least *one* occurrence $[S_i, p]$ of w such that all remaining three occurrences of w have positive scores when compared to $[S_i, p]$. To sample a list of homologous P-blocks, we randomly select spaced-word occurrences with respect to P from the input sequences and their reverse complements. For each selected $[S_i, p]$, we then randomly select occurrences of the same spaced word from sequences $S_j \neq S_i$, until we have found three occurrences of w from three different sequences that all have positive scores with $[S_i, p]$.

To find spaced-word matches efficiently we first sort the list of all occurrences of spaced words with respect to P in lexicographic order. This way, we obtain a list of spaced-word occurrences where all occurrences of the same spaced word w are appearing as a contiguous block. Once we have sampled a homologous P-block as described, we remove the four occurrences of w from our list of spaced-word occurrences, so no two of the sampled P-blocks can contain the same occurrence of a spaced word. The algorithm continues to sample P-blocks until no further P-blocks can be found, or until a given number of P-blocks is reached. By default, *Multi-SpaM* uses a maximal number of $M = 1,000,000$ P-blocks, but this parameter can be adjusted by the user.

For each of the sampled quartet P-blocks, we infer an unrooted tree topology. This most basic *unrooted* phylogenetic unit is called a *quartet* topology; there are three possible different quartet topologies for a set of four taxa. To identify the best of these three topologies, we use the *Maximum Likelihood* program *RAxML* [51]. We note that *RAxML* is a general *Maximum-Likelihood* software, its use in our context is fairly degenerated, as we only use it to infer optimal quartet topologies.

Figure 1: Example of a quartet tree topology.

After having the optimal tree topology for each of the sampled quartet P-blocks, we need to amalgamate them into a single tree spanning the entire taxa set. This task is denoted the *Supertree Task* [6] and is known to be *NP hard*, even for the special case where the input is limited to quartets topologies, as in our case [52]. Nevertheless there are several heuristics for this task, with *MRP* [4, 41] the most popular. Here we chose to use *Quartet MaxCut* [47, 48] that proved to be faster and more accurate for this kind of input [2]. In brief, *Quartet MaxCut* partitions recursively the taxa set where each such partition corresponds to a split in the final tree. In each such recursive step, a graph over the taxa set is built where the set of quartets induces the edge set in that graph. The idea is to partition the vertex set (the taxa) such that the minimum quartets are violated. This is achieved by a *semidefinite-programming*-like algorithm that embeds the graph on the unit sphere and applies a random hyperplane through the sphere.

- document style
- baselineskip
- front matter
- keywords and MSC codes
- theorems, definitions and proofs
- lables of enumerations
- citation style and labeling.

3. Front matter

The author names and affiliations could be formatted in two ways:

(1) Group the authors per affiliation.

(2) Use footnotes to indicate the affiliations.

See the front matter of this document for examples. You are recommended to conform your choice to the journal you are submitting to.

4. Bibliography styles

There are various bibliography styles available. You can select the style of your choice in the preamble of this document. These styles are Elsevier styles based on standard styles like Harvard and Vancouver. Please use BibTEX to generate your bibliography and include DOIs whenever available.

Here are two sample references: [**? ?**].

References

References

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

[2] Eliran Avni, Zahi Yona, Reuven Cohen, and Sagi Snir. The performance of two supertree schemes compared using synthetic and real data quartet input. *Journal of Molecular Evolution*, 86:150–165, 2018.

[3] Lorraine A.K. Ayad, Panagiotis Charalampopoulos, Costas S. Iliopoulos, and Solon P. Pissis. Longest common prefixes with k-errors and applications. *arXiv:1801.04425 [cs.DS]*, 2018.

[4] B.R. Baum. Combining trees as a way of combining data sets for phylogenetic inference. *Taxon*, 41:3–10, 1992.

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

[6] O.R.P. Bininda-Emonds. *Phylogenetic supertrees: Combining information to reveal the Tree of Life*. Computational Biology. Springer, 2004.

[7] Raquel Bromberg, Nick V. Grishin, and Zbyszek Otwinowski. Phylogeny reconstruction with alignment-free method that corrects for horizontal gene transfer. *PLOS Comput Biol*, 12:e1004985, 2016.

[8] Giuseppe Cattaneo, Umberto Ferraro Petrillo, Raffaele Giancarlo, and Gianluca Roscigno. An effective extension of the applicability of alignment-free biological sequence comparison algorithms with Hadoop. *The Journal of Supercomputing*, 73:1467–1483, 2017.

[9] Maria Chatzou, Evan W. Floden, Paolo Di Tommaso, Olivier Gascuel, and Cedric Notredame. Generalized bootstrap supports for phylogenetic analyses of protein sequences incorporating alignment uncertainty. *Systematic Biology*, page syx096, 2018.

[10] 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, 2002.

[11] Benny Chor and Tamir Tuller. Maximum likelihood of evolutionary trees is hard. In Satoru Miyano, Jill Mesirov, Simon Kasif, Sorin Istrail, Pavel A. Pevzner, and Michael Waterman, editors, *Research in Computational Molecular Biology*, pages 296–310, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[12] Matteo Comin and Michele Schimd. Assembly-free genome comparison

based on next-generation sequencing reads and variable length patterns. *BMC Bioinformatics*, 15:S1, 2014.

[13] Daniel A. Dalquen, Maria Anisimova, Gaston H. Gonnet, and Christophe Dessimoz. ALF - a simulation framework for genome evolution. *Molecular Biology and Evolution*, 29:1115–1123, 2012.

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

[15] Joseph Felsenstein. Evolutionary trees from DNA sequences:a maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981.

[16] Joseph Felsenstein. Distance methods for inferring phylogenies: a justification. *Evolution; international journal of organic evolution*, 38:16–24, 1984.

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

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

[19] D. F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 25:351–360, 1987.

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

[21] L.R. Foulds and R.L Graham. The steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics*, 3:43–49, 1982.

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

[23] Samuele Girotto, Matteo Comin, and Cinzia Pizzi. FSH: fast spaced seed hashing exploiting adjacent hashes. *Algorithms for Molecular Biology*, 13:8, 2018.

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

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

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

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

[28] 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.

[29] 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.

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

[31] Chris-Andre Leimeister, Jendrik Schellhorn, Svenja Schöbel, Michael Gerth, Christoph Bleidorn, and Burkhard Morgenstern. Prot-spam:

Fast alignment-free phylogeny reconstruction based on whole-proteome sequences. *bioRxiv*, 2018.

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

[33] 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.

[34] 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.

[35] R.J. Newton, L.E. Griffin, K.M. Bowles, C. Meile, S. Gifford, C.E. Givens, E.C. Howard, E. King, C.A. Oakley, C.R. Reisch, J.M. Rinta-Kanto, S. Sharma, S. Sun, V. Varaljay, M. Vila-Costa, J.R. Westrich, and M.A. Moran. Genome characteristics of a generalist marine bacterial lineage. *The ISME Journal*, 4:784–798, 2010.

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

[37] OpenMP Forum. OpenMP C and C++ Application Program Interface, Version 2.0. `http://www.openmp.org`. Technical report, March 2002.

[38] Rachid Ounit and Stefano Lonardi. *Algorithms in Bioinformatics: 15th International Workshop, WABI 2015, Atlanta, GA, USA, September 10-12, 2015, Proceedings*, chapter Higher Classification Accuracy of Short Metagenomic Reads by Discriminative Spaced *k*-mers, pages 286–295. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[39] Umberto Ferraro Petrillo, Concettina Guerra, and Cinzia Pizzi. A new distributed alignment-free approach to compare whole proteomes. *Theoretical Computer Science*, 698:100–112, 2017.

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

[41] M.A. Ragan. Matrix representation in reconstructing phylogenetic-relationships among the eukaryotes. *Biosystems*, 28:47–55, 1992.

[42] Jie Ren, Xin Bai, Yang Young Lu, Kujin Tang, Ying Wang, Gesine Reinert, and Fengzhu Sun. Alignment-free sequence analysis and applications. *arXiv:1803.09727[q-bio.QM]*, 2018.

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

[44] Tanmoy Roychowdhury, Anchal Vishnoi, and Alok Bhattacharya. Next-generation anchor based phylogeny (nexabp): Constructing phylogeny from next-generation sequencing data. *Scientific Reports*, 3:2634, 2013.

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

[46] Fabian Sievers, Andreas Wilm, David Dineen, Toby J Gibson, Kevin Karplus, Weizhong Li, Rodrigo Lopez, Hamish McWilliam, Michael Remmert, Johannes Söding, Julie D Thompson, and Desmond G Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, 7:539, 2011.

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

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

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

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

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

[52] M. Steel. The complexity of reconstructing trees from qualitative characters and subtress. *Journal of Classification*, 9:91–116, 1992.

[53] S. V. Thankachan, S. P. Chockalingam, Y. Liu, A. Krishnan, and S. Aluru. A greedy alignment-free distance estimator for phylogenetic inference. *BMC Bioinformatics*, 18:238, 2017.

[54] Sharma V. Thankachan, Alberto Apostolico, and Srinivas Aluru. A provably efficient algorithm for the k-mismatch average common substring problem. *Journal of Computational Biology*, 23:472–482, 2016.

[55] 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.

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

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