Syst. Biol. 70(2):236–257, 2021

DOI:10.1093/sysbio/syaa050
Advance Access publication July 12, 2020

The Cumulative Indel Model: Fast and Accurate Statistical Evolutionary Alignment

NICOLA DE MAIO*

European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton CB10 1SD, UK
**Correspondence to be sent to: European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton CB10 1SD, UK; E-mail: demaio@ebi.ac.uk.*

Received 21 March 2019; reviews returned 21 June 2020; accepted 23 June 2020
Associate editor: Edward Susko

Abstract.—Sequence alignment is essential for phylogenetic and molecular evolution inference, as well as in many other areas of bioinformatics and evolutionary biology. Inaccurate alignments can lead to severe biases in most downstream statistical analyses. Statistical alignment based on probabilistic models of sequence evolution addresses these issues by replacing heuristic score functions with evolutionary model-based probabilities. However, score-based aligners and fixed-alignment phylogenetic approaches are still more prevalent than methods based on evolutionary indel models, mostly due to computational convenience. Here, I present new techniques for improving the accuracy and speed of statistical evolutionary alignment. The "cumulative indel model" approximates realistic evolutionary indel dynamics using differential equations. "Adaptive banding" reduces the computational demand of most alignment algorithms without requiring prior knowledge of divergence levels or pseudo-optimal alignments. Using simulations, I show that these methods lead to fast and accurate pairwise alignment inference. Also, I show that it is possible, with these methods, to align and infer evolutionary parameters from a single long synteny block (≈530 kbp) between the human and chimp genomes. The cumulative indel model and adaptive banding can therefore improve the performance of alignment and phylogenetic methods. [Evolutionary alignment; pairHMM; sequence evolution; statistical alignment; statistical genetics.]

Efficient and accurate modeling of sequence evolution, including substitutions and indels (insertions and deletions), is key for accurate phylogenetic, selection, and alignment inference, but so far a simple solution to this problem has remained elusive (Miklós et al. 2009). Phylogenetic and molecular evolution methods usually analyze a fixed alignment inferred from a multiple sequence aligner and model only substitution events while treating gaps in the alignment as missing data (Yang and Rannala 2012). On the other hand, multiple sequence aligners are typically heuristic and score-based, sometimes using fixed phylogenies (guide trees, Notredame 2007). However, evolutionary modeling of indels is important for improving alignment inference and reducing biases in molecular evolution analyses (Löytynoja and Goldman 2008b). Furthermore, statistical and evolutionary modeling of indels allows parameter estimation (Lunter 2007b), unbiased alignment sampling (Metzler et al. 2001; Metzler 2003), joint alignment-phylogeny inference (Mitchison and Durbin 1995; Mitchison 1999; Holmes and Bruno 2001; Lunter et al. 2005; Fleissner et al. 2005; Novák et al. 2008; Redelings and Suchard 2005; Bouchard-Côté et al. 2009; Westesson et al. 2012), and realistic simulations (Cartwright 2005; Rosenberg 2005; Fletcher and Yang 2009; Strope et al. 2009).

Some evolutionary indel models are similar to phylogenetic substitution models (McGuire et al. 2001; Rivas and Eddy 2008), with the limitation that alignment columns are considered independent. One of the most popular evolutionary indel models, TKF91 (Thorne et al. 1991), has a similar limitation, in that it models indels as 1-residue events. This limitation was removed in the TKF92 model (Thorne et al. 1992). The TKF91 and TKF92 models can be represented as Hidden Markov Models (Hein 2000; Holmes and Bruno 2001), which allows their efficient and flexible use in a variety of settings. Pair Hidden Markov Models ("pairHMMs") have in fact long been used in statistical alignment although originally not in the context of evolutionary indel models (Durbin et al. 1998). Recently developed pairHMMs, however, do consider the indel evolutionary process, similarly to TKF92, to account for the effect of different expected alignment patterns at different levels of sequence divergence (Löytynoja and Goldman 2005; Redelings and Suchard 2007).

While the TKF92 model and other similar pairHMMs have been the predominant models in statistical alignment, in recent years there has been considerable effort in attempting to increase the realism of evolutionary indel models; in particular, one target for improvement has been the relaxation of the TKF92 assumption that sequences are composed of unsplittable "fragments" of geometrically distributed lengths. The most popular of these new models is the "long indel model" (Miklós et al. 2004; Levy Karin et al. 2019) that is based on assumptions generally considered as realistic. The long indel model does not assume unsplittable fragments and allows multi-residue instantaneous indels. The long indel model assumes that at each instant, any stretch of contiguous residues can be deleted from a sequence, and any stretch of residues can be inserted in any position of the sequence. Despite the positive qualities of the long indel model, its applications have remained limited, partly due to its complexity, and partly to the computational demand of its implementations.

Here, I present the "cumulative indel model," a new evolutionary pairHMM that closely approximates the features of the long indel model, additionally

assuming geometrically distributed indel lengths, and without assuming reversibility. In the following, for brevity, I will refer to the model being approximated as simply the "general indel model". Given evolutionary parameters such as divergence time and insertion and deletion rates, the cumulative indel model accurately predicts expected numbers and lengths of stretches of gap columns in an alignment generated under the general indel model. Unlike the long indel model, the cumulative indel model assumes that stretches of gap columns in an alignment have geometrically distributed lengths. While this assumption is generally not correct, it allows implementation of the cumulative indel model as a classical finite pairHMM, granting simplicity, computational efficiency, and applicability.

I also introduce a new dynamic programming technique called "adaptive banding," which can drastically reduce time and memory demands of pairHMMs, as well as of classical alignment algorithms. Unlike previous related techniques (Chao et al. 1992; Hein et al. 2000; Havgaard et al. 2007; Westesson et al. 2012; Bogusz and Whelan 2017), adaptive banding is flexible and accurate, being suitable for any alignment without requiring prior knowledge of divergence level or pseudo-optimal alignments.

Using simulations, I compare the cumulative indel model with previous pairHMM evolutionary indel models. I performed all simulations under the general indel model, rather than under the cumulative indel model proposed here. I consider the problem of inferring pairwise alignments as well as inferring evolutionary parameters from homologous sequence pairs. Parameter estimation is useful not only to improve alignment accuracy, but also to infer and compare the frequency of different mutational events and the effects of selective forces on genome evolution. I show that the cumulative indel model offers considerable advantages, such as high alignment accuracy and parameter interpretability. Furthermore, I compare classical score-based pairwise alignment techniques with pairHMMs, and show that adaptive banding allows accurate and efficient genome-wise alignment and parameter inference for closely related sequences.

Materials and Methods

Definition of Alignment

A residue in a sequence is considered homologous to a residue in another sequence if they are descended from the same ancestral residue, possibly through substitution but not insertion. An evolutionary pairwise alignment (from now on just "alignment") states the homology relationships between the residues of two sequences: homologous residues are in the same alignment column, while non-homologous residues are in different columns. For simplicity, I restrict consideration to the alignment of an ancestral and a descendant sequence, so that the three types of alignment columns correspond to homologous residues

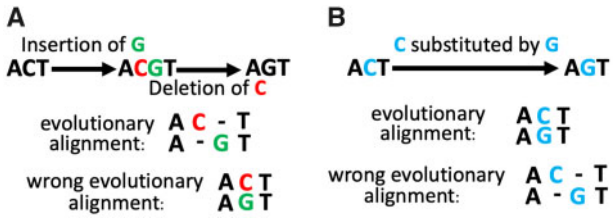

FIGURE 1. Example of correct and wrong evolutionary alignments. a) An ancestral "ACT" sequence undergoes an insertion of a "G" and the deletion of the "C". b) An ancestral "ACT" sequence undergoes a substitution of the "C" into a "G". The correct evolutionary alignments in each scenario of the ancestral ("ACT") and derived ("AGT") sequences are in the center. Wrong evolutionary alignments are in the bottom. Inserted bases are colored in green, deleted bases in red, and substitutions in blue.

(match columns) inserted residues (insertion columns) and deleted residues (deletion columns). A "match" alignment column (often denoted simply as "M") contains residues for both sequences, and infers that these residues are homologous, and so did not undergo insertion or deletion within the considered evolutionary history separating the two sequences. An inferred alignment can be wrong if homologous residues are placed in different columns, or if non-homologous residues are placed in the same column. A "deletion" alignment column (or "D") contains a residue only for one of the two sequences aligned, and in particular the sequence that is considered ancestor of the other; a gap character "-" appears instead in place of the residue of the descendant sequence. Only residues that are in the ancestral sequence can be in a D column, while residues that are first inserted and then later on deleted (and so do not appear in either of the two sequences) are not present in the alignment. The sum of the numbers of M and D columns is therefore equal to the length of the ancestral sequence. "Insertion" alignment columns (or "I") contain a residue only for the descendant sequence, and they contain a gap character in place of the ancestral residue. The sum of the numbers of M and I columns in an alignment equals the length of the descendant sequence. In Figure 1, I show examples of correct and wrong evolutionary alignments: correctness does not depend only on the two considered sequences, but also on the (usually unobserved) indel events generating them from their common ancestor.

Often, multiple evolutionary alignments can be correct at the same time (Fig. 2). This can happen if some alignment columns are "exchangeable," that is, if inverting their order in the alignment does not affect the homology statements of the alignment (like the two central columns in the alignments in Fig. 2). Following (Miklós et al., 2004), I denote a class of exchangeable columns (all I and D columns between two consecutive M columns) as a"chop zone." A chop zone can be represented by many different alignments, which I call "chop zone alignments." For example, the alignments in Figure 2 are different chop zone alignments, but they all represent the same chop zone. In the probabilistic sequence evolution models TKF91 and TKF92, only one

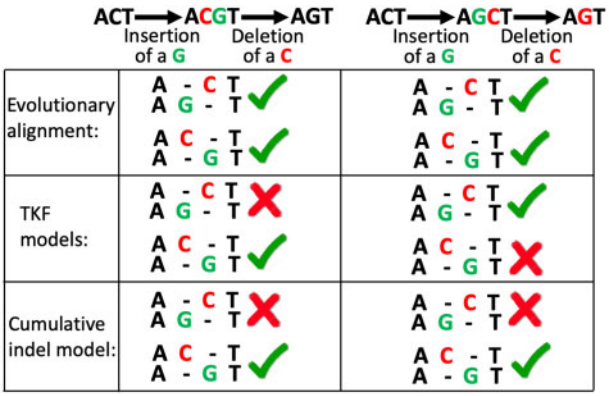

FIGURE 2. Different definitions of evolutionary alignment. Above the matrix: two evolutionary histories with the same ancestral ("ACT") and derived ("AGT") sequences, but with inverted order of the "G" insertion and "C" deletion. Matrix top row: each history has two correct evolutionary alignments. Middle row: the TKF models have one correct alignment per indel history, and in this case the correct alignments differ from each other. This is because in the TKF models inserted residues are thought as generated by "links" near extant residues (see Thorne et al. 1991, 1992 for details). Bottom row: in the cumulative indel model, only one alignment is consistent with both indel histories. Inserted bases are colored in green, deleted ones in red.

chop zone alignment is consistent with a history of indel events (Fig. 2 and Supplementary Fig. S1 available on Dryad at http://dx.doi.org/10.5061/dryad.rbnzs7h8m; see Thorne et al. 1991, 1992 for details), so different alignments of the same chop zone make different predictions about the evolutionary history that generated that chop zone. If the probability of an alignment is defined as the sum of the probabilities of the evolutionary histories consistent with that alignment, then it conveniently follows that under the TKF91 and TKF92 models all alignment probabilities sum to one (i.e., different alignments share no evolutionary histories). Other pairHMMs are more loosely defined, and while different chop zone alignments can have different probabilities, the order of I and D column in a chop zone alignment does not have a clear interpretation (see e.g., Löytynoja and Goldman 2005; Redelings and Suchard 2007). In the context of my model, I only assign nonzero probability to (i.e., I consider valid) alignments where, when an insertion and a deletion column are exchangeable, the deletion column appears on the left end of the insertion column (Fig. 2 and Supplementary Fig. S1 available on Dryad). I define the probability of a valid alignment as the sum of the probabilities of all evolutionary histories that result in the same sequences and homology relationships. This way, the probability that two different alignments are both true is always 0. Note that this restriction on alignment column is only made during inference under my model, and not during simulations, which are performed under the general indel model adhering to the TKF92 convention regarding chop zone alignments. When assessing the accuracy of alignment inference, only the correctness of the homology statements is then taken into account, and not the order of columns within a chop zone. I will refer to a group of contiguous insertion columns as a "cumulative insertion," and to a group of consecutive deletion columns as a "cumulative deletion." As a cumulative insertion refers to a group of alignment columns, and therefore to homology statements, it should not be confused with individual instantaneous insertion events, which are modeled but not inferred or represented in an alignment. Unlike the TKF91 and TKF92 models, there is a one-to-one identity between valid alignments and valid sets of homology statements, so that the alignment space that needs to be explored during alignment inference is smaller. Restricting the space of allowed alignments can therefore reduce the computational demand of statistical inference, but can, potentially, also reduce the ideal accuracy of the model, as I later discuss more in detail. In Supplementary Section S3 available on Dryad, I extend the definition of alignments allowed by the cumulative indel model to the case of a multiple sequence alignments associated with a phylogenetic tree.

The General Indel Model

Here, I briefly describe the model that I use in most simulations, and that I refer to as the "general indel model" (GIM). The GIM is a specific instance of the model implemented in simulators such as INDELible (Fletcher and Yang 2009), where I additionally restrict to the case that indel events have geometrically distributed lengths. The GIM is also similar to the Long Indel Model (Miklós et al. 2004), but here I specifically assume geometrically distributed indel lengths, while I do not necessarily assume reversibility, and I do not assume upper bounds to indel lengths or numbers of indel events. The GIM is not the most general indel model that can be used for inference or simulations (in particular due to the assumption of geometrically distributed indel lengths), and can be distinguished from the model proposed for inference in this manuscript (the Cumulative Indel Model) because the latter involves several simplifying approximations to reduce computational complexity. One of the aims of the Cumulative Indel Model is to closely (but approximately) predict the features of alignments generated under the GIM.

The GIM assumes that evolution occurs in continuous time, and that given the current sequence s, its future evolution depends on s but not on the past evolutionary history of s (sequence evolution is a continuous time Markov process). Three types of mutation events cause sequence evolution. Substitutions can instantaneously replace one character in s with another character; as typical, I assume that the substitution rate of a character does not depend on its position along s or on the other characters in s. More specifically, in the following I will always employ the HKY85 model of substitutions (Hasegawa et al. 1985). I will also assume that time is measured in units of expected substitutions per site,

so that the substitution rate per site is constantly 1.0 across time. The other two types of mutational events are insertions and deletions. Insertion events can happen at any position in the sequence (between any two characters, before the first character in s, or after the last character in s) with the same rate r_i constant through time and along the sequence. When an insertion happens, it adds a stretch of characters to s at the considered position. I assume that the length n of the stretch of character is taken from a geometric distribution with parameter g_i, that is, the probability of length n is $(g_i)^{n-1}(1-g_i)$. The insertion rate is not affected by the specific sequence of characters of s, however, the longer is s the more insertions are expected overall in s per unit of time. The specific characters added to s with an insertion are sampled independently of one another from the equilibrium distribution of the substitution process (in this case, from the HKY85 model). Deletions remove a continuous stretch of characters from s. A deletion of length n starts at a certain character in s and removes that character from s as well as the following $n-1$ characters. The rate r_d at which a deletion happens starting at a certain position of s is constant through time and through s and is not affected by the specific sequence of characters in s (however, again, the longer s the more deletions are expected overall per unit of time). The length n of a deleted stretch is also sampled from a geometric distribution with parameter g_d, that is, the probability of length n is $(g_d)^{n-1}(1-g_d)$. Note that, under these assumptions, the GIM is not necessarily a reversible model of sequence evolution. Defining the deletion process near the edges of s is not straightforward. Here, we assume that s is embedded within an infinite sequence, so that the initial characters in s can also be deleted due to deletion events starting before s (see also Miklós et al. 2004; Cartwright 2005; Fletcher and Yang 2009). While I assume geometric length distributions for indel events, power law indel length distributions have sometimes been recommended as more realistic (Cartwright 2008). Geometric indel lengths have however considerable computational advantages as they can be naturally translated into pairHMM parameters.

The Cumulative Indel Model

Here, I aim to accurately and efficiently approximate features of the GIM, such as the distributions of cumulative indels expected between ancestral sequence s_1 and descendant s_2 separated by a divergence time t. Accurate predictions of the distributions of indel columns in an alignment will then be used, in later sections, to calculate the probability of an alignment, and, more importantly, to infer alignments and evolutionary parameters within a pairHMM. The model proposed here is not in itself a model of sequence evolution, and is not meant to simulate the evolution of sequence; it instead defines a pairHMM, which can be used to describe the probabilities of alignments, to perform statistical inference of alignments or over alignments, and possibly to simulate alignments. However, it in general makes more sense to simulate alignments under the exact GIM, as I do here, and use the cumulative indel pairHMM instead to perform statistical inference from the simulated sequences.

Combining accuracy and efficiency in statistical alignment is still an open problem (Holmes 2017). To derive useful transition probabilities in the final pairHMM, I assume, as an approximation, that at any time t cumulative indel lengths are geometrically distributed with parameters g_i^t for insertions and g_d^t for deletions. Geometric distributions for cumulative indel lengths have also the advantage of being maximum entropy distributions given fixed expected lengths for cumulative indels and are therefore a convenient and principled choice given that the true distributions are not known. Such geometric distributions are expected to be a good approximation for short divergence times t; however, the dynamics of indel distributions under the GIM are complex, and at divergence time $t>0$ cumulative indels are not guaranteed to be geometrically distributed even if instantaneous indel event lengths are (Fig. 5F, Supplementary Fig. S3 available on Dryad, and Rivas and Eddy 2015). Therefore, the assumption I make of geometrically distributed cumulative indel lengths is not consistent with the other assumptions of the GIM, and, as such, estimation under this model is not expected to be statistically consistent under the GIM. The pairHMM obtained from this approximation is intended as an approximation of the GIM providing a computationally efficient mean for calculating approximate alignment probabilities. Another approximating assumption I make is that the non-empty cumulative insertion (respectively, deletion) length distribution is independent of the presence and length of a non-empty cumulative deletion (respectively, insertion) in the same chop zone (the part of an alignment between two consecutive match columns). This means that the model does not take into account that, at high divergence, cumulative indel length, and cumulative insertion length within the same chop zone can be correlated.

For most of the following, unless stated otherwise, I assume that sequences s_1 and s_2 are parts of infinite genomes; the considered true alignment of s_1 and s_2 is a selected stretch of the true infinite alignment of the two infinite genomes, with uniform (improper) distributions over position and length of the selection. This means assuming a uniform prior measure (improper probability) over alignment length. Alternative assumptions are discussed in Supplementary Section S4 available on Dryad. Unlike TKF91 and TKF92, I do not assume sequence length equilibrium or reversibility of sequence evolution, and I do not assume that the prior distribution over ancestral sequence length is affected by the parameters of the indel process.

TABLE 1. Parameters and variables of the cumulative indel model.

Parameters	Description	Human-chimp comparison	Human-chimp ×100	Max divergence simulated
t	Divergence time between ancestral and descendant sequence	0.0118	1.176	1.0
r_i	Instantaneous indel rate	0.0657	0.0657	0.5
r_d	Instantaneous deletion rate	0.0657	0.0657	0.5
g_i	Instantaneous insertion length geometric distribution parameter	0.759	0.759	0.75
g_d	Instantaneous deletion length geometric distribution parameter	0.759	0.759	0.75
Variables				
g_i^t	Cumulative insertion length geometric distribution parameter	0.759	0.791	0.925
g_d^t	Cumulative deletion length geometric distribution parameter	0.759	0.791	0.923
P_m^t	Probability that an ancestral residue is extant	0.9968	0.726	0.136
A_i^t	Probability that an ancestral residue is extant and followed by a nonempty cumulative insertion	7.7×10^{-4}	0.0572	0.0650
P_i^t	Probability that an extant ancestral residue is followed by a nonempty cumulative insertion	7.72×10^{-4}	0.0788	0.479
A_d^t	Probability that an ancestral residue is extant and followed by a nonempty cumulative deletion	7.7×10^{-4}	0.0574	0.0669
P_d^t	Probability that an extant ancestral residue is followed by a nonempty cumulative deletion	7.72×10^{-4}	0.0791	0.493
L_i^t	Expected inserted residues per ancestral residue	0.0032	0.274	0.864
A_{id}^t	Probability that an ancestral residue is extant and followed by both a nonempty cumulative insertion and deletion	1.696×10^{-6}	0.0109	0.0464
P_{id}^t	Probability that an extant ancestral residues is followed by both a nonempty cumulative insertion and deletion	1.701×10^{-6}	0.0151	0.342

The Differential Equations of Cumulative Indel Distributions

To keep track of the evolution over time t of the distributions of cumulative indels, here I define a set of variables of the alignment. These variables track the frequency of nonempty cumulative indels in the alignment and their average lengths. Due to the assumption of geometrically distributed cumulative indel lengths, these variables are sufficient to approximate alignment probability. All parameters and variables used here are listed and described in Table 1, including example values estimated from the human-chimp data set considered below ("Results" section), for the same scenario but with 100-fold longer divergence time t between the two species, and values for the simulations with the highest levels of divergence considered ($t=1.0$ and $r=0.5$, see "Simulations" section). These variables define the pairHMM described below, and as such, they are used to calculate the probability (or measure) of alignments.

The first variable, P_m^t is the probability that an ancestral residue is still present in the descendant sequence (it has not been deleted) after time t. As initially ($t=0$) no residue is deleted, one has $P_m^0=1$. Since r_d is the deletion rate, $1/(1-g_d)$ is the average instantaneous deletion length, and all residues are deleted at the same rate, extant ancestral residues are deleted at rate $r_d/(1-g_d)$. It follows that:

$$\frac{dP_m^t}{dt}=-r_dP_m^t/(1-g_d). \quad (1)$$

Another way to derive this result is by considering that for a very short time δ one has that the probability that an extant ancestral residue is deleted is

$$\delta r_d\sum_{h\geq0}\sum_{j\geq h+1}g_d^{j-1}(1-g_d)+o(\delta) \quad (2)$$

here, δ is small enough so that one can ignore the probability of more indels affecting the same region, and δr_d approximates the probability of a deletion event starting at any position; parameter $h\geq0$ represents the starting position of a deletion (in terms of bases at the left of the considered extant ancestral residue); $g_d^{j-1}(1-g_d)$ is the probability that the considered deletion is long enough to delete the considered extant ancestral residue. Since $\sum_{h\geq0}\sum_{j\geq h+1}g_d^{j-1}(1-g_d)=1/(1-g_d)$, Equation 1 follows. The solution to Equation 1 is:

$$P_m^t=P_m^0e^{-tr_d/(1-g_d)}=e^{-tr_d/(1-g_d)}. \quad (3)$$

A second variable I consider is L_i^t, the expected number of inserted residues per ancestral residue. The initial value is $L_i^0=0$. Inserted residues are deleted at rate $r_d/(1-g_d)$, as above for ancestral residues. New residues are instead inserted at rate $(P_m^t+L_i^t)r_i/(1-g_i)$ since $1/(1-g_i)$ is the expected instantaneous insertion length, and since insertions can appear next to either extant ancestral residues or inserted residues. Combining the rates of these two types of events one obtains:

$$\frac{dL_i^t}{dt}=(P_m^t+L_i^t)r_i/(1-g_i)-L_i^tr_d/(1-g_d). \quad (4)$$

Substituting the function for P_m^t from equation 3 and considering the initial condition of $L_i^0=0$, one finds that the solution is:

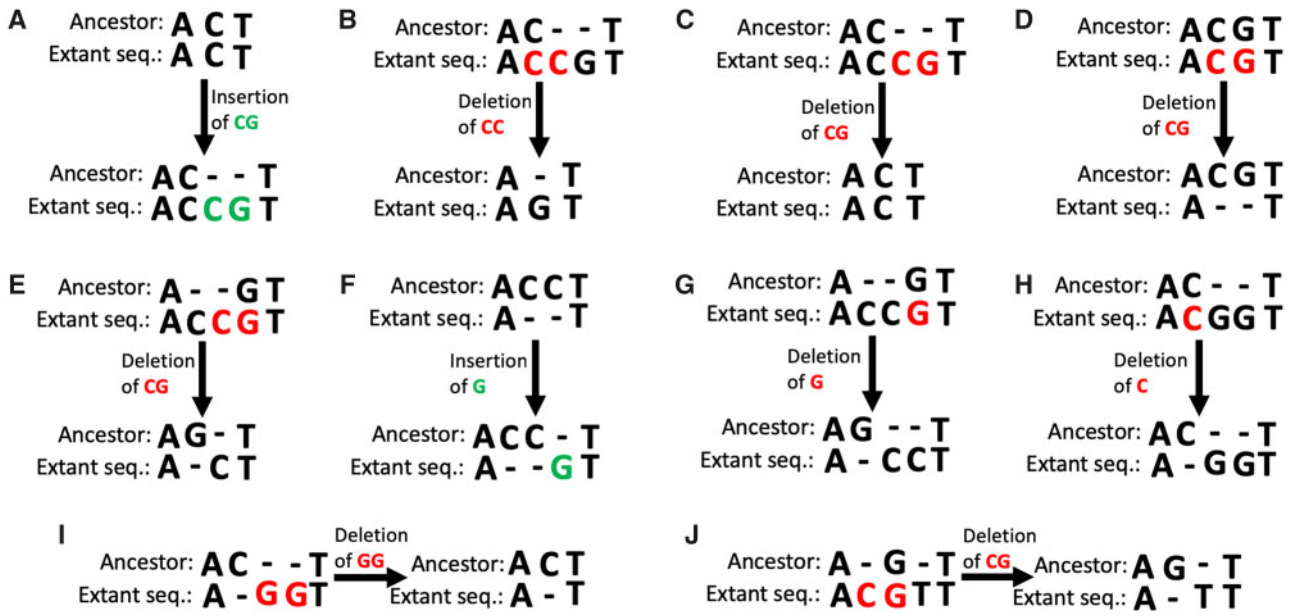

FIGURE 3. Events affecting cumulative indel dynamics. Each sub-plot represents a type of event that can affect a variable of the cumulative indel model. In each sub-plot, the alignment between the ancestor sequence and the extant sequence before the event is shown, followed (after the arrow that represents the indel event) by the alignment of the ancestor sequence and the extant sequence after the event. Inserted bases are colored in green, deleted ones in red.

$$L_i^t = e^{(r_i/(1-g_i)-r_d/(1-g_d))t}(1-e^{tr_i/(1-g_i)}) = P_m^t(e^{tr_i/(1-g_i)}-1). \quad (5)$$

In deriving P_m^t and L_i^t above, I have not used the assumption of geometrically distributed cumulative indels lengths or any other approximations to the GIM. As such, Equations 3 and 5 hold exactly for the GIM. For all other variables below, I use instead the assumption that cumulative indels have geometrically distributed lengths, and often other approximations such as neglecting low-probability events. As such, all the remaining variables below in this section need to be treated as approximations to the GIM. These approximations are expected to be accurate for short divergence times (or equivalently low indel rates) when most cumulative indels are made of individual indel events; their quality is instead expected to degrade at very high divergence. The derivatives of these remaining variables in the model define a system of ordinary differential equations, which I solve numerically.

A third variable A_i^t is the probability that an ancestral residue is extant (not deleted) and followed by a non-empty cumulative insertion. The initial value is $A_i^0 = 0$. A related variable that I also define here for convenience is P_i^t, the probability that an extant ancestral residue is followed by a non-empty cumulative insertion. The difference between A_i^t and P_i^t is that the former represents a proportion over all ancestral residues (also deleted ones), while the latter is the proportion over only extant residues, and one therefore has $A_i^t = P_i^t P_m^t$. Variable A_i^t is affected by four types of events. New insertions can appear right after match columns previously not followed by insertions, so increasing A_i^t with rate $(P_m^t - A_i^t)r_i$ (see e.g., Fig. 3a). Deletions can remove match columns followed by non-empty cumulative insertions, decreasing A_i^t at rate $A_i^t r_d/(1-g_d)$. Deletions can start right after a match column not followed by insertions, and remove a match column followed by a non-empty cumulative insertion, but not delete the whole cumulative insertion (see e.g., Fig. 3b); this causes A_i^t to increase at rate $(P_m^t - A_i^t)r_d P_i^t(1-g_d)/((1-g_d(1-P_i^t))(1-g_d g_i^t))$, since $P_i^t/(1-g_d(1-P_i^t))$ is the probability that a deletion reaches a nonempty cumulative insertion, and $(1-g_d)/(1-g_d g_i^t)$ is the probability that the cumulative insertion is not completely deleted. Here and below, I neglect the contributions of deletions removing multiple entire cumulative insertions, as these events are expected to be relatively rare unless deletions events can be considerably longer than insertion events. In large-scale alignments, such as genome-level alignments, such large deletions events can be particularly plausible (see e.g., Gregory 2004; Newman et al. 2005), and so this approximation could be a target for future improvement of the model. Lastly, deletions can remove whole cumulative insertions after a non-deleted match column (see e.g., Fig. 3C), therefore decreasing A_i^t at rate $A_i^t r_d(1-g_i^t)(1-g_d)/((1-g_d(1-P_i^t))(1-g_d g_i^t))$: here one has to require that the deletion removes the whole cumulative insertion, which has probability $(1-g_i^t)/(1-g_d g_i^t)$, and that the deletion does not continue into

another cumulative insertion, with probability $(1-g_d)/(1-g_d(1-P_i^t))$. Here, and below, I use the assumption that at any point in time non-empty cumulative indels are geometrically distributed, and as such, while these derivatives usually provide a good approximation of evolutionary dynamics under a GIM, they do not define in themselves a self-consistent mathematical model of sequence evolution. Combining these four types of events leads to:

$$\begin{aligned}\frac{dA_i^t}{dt} &= (P_m^t - A_i^t)r_i - A_i^t r_d/(1-g_d) + \\ &+ (P_m^t - A_i^t)r_d \frac{P_i^t(1-g_d)}{(1-g_d(1-P_i^t))(1-g_d g_i^t)} + \\ &- A_i^t r_d \frac{(1-g_i^t)(1-g_d)}{(1-g_d(1-P_i^t))(1-g_d g_i^t)}. \end{aligned} \quad (6)$$

It is possible to express g_i^t in terms of L_i^t and P_i^t (and therefore A_i^t) since

$$L_i^t = P_m^t P_i^t/(1-g_i^t). \quad (7)$$

The second variable I consider in the system of differential equations is A_d^t, the probability that an ancestral residue is extant and followed by a nonempty cumulative deletion. This has initial value $A_d^0 = 0$. Another variable that I also use for notational convenience is P_d^t, the probability that an extant ancestral residue is followed by a non-empty cumulative deletion. This means $A_d^t = P_d^t P_m^t$. Variable A_d^t is subject to three types of events. Deletions that start at (and include) a match column preceded on the left by an empty cumulative deletion (see e.g., Fig. 3D) create new nonempty cumulative deletions at rate $(P_m^t - A_d^t)r_d$. Deletions remove match columns followed by nonempty cumulative deletions, and therefore reduce A_d^t at rate $A_d^t r_d/(1-g_d)$. Lastly, deletions starting at an inserted residue can create a new nonempty cumulative deletion (see e.g., Fig. 3E) at rate $(P_i^t - P_{id}^t)L_i^t r_d g_d(1-g_i^t)/(P_i^t(1-g_d g_i^t))$. Summing these contributions one obtains:

$$\begin{aligned}\frac{dA_d^t}{dt} &= (P_m^t - A_d^t)r_d - A_d^t r_d/(1-g_d) \\ &+ \frac{P_i^t - P_{id}^t}{P_i^t} \frac{L_i^t r_d g_d (1-g_i^t)}{1-g_d g_i^t}. \end{aligned} \quad (8)$$

Because at time t one has a proportion $1-P_m^t$ of deleted ancestral residues, the average length of a nonempty cumulative deletion is $(1-P_m^t)/A_d^t$; however, due to the assumption of geometric length cumulative deletions, this quantity is also equal to $1/(1-g_d^t)$. It follows that one can express g_d^t in terms of A_d^t:

$$g_d^t = 1 - \frac{A_d^t}{1-P_m^t}. \quad (9)$$

The third and last variable whose derivative is part of the system of differential equations is A_{id}^t, the probability that an ancestral residue is extant and followed by a non-empty cumulative insertion and a non-empty cumulative deletion. The initial value is $A_{id}^t = 0$. As usual, for convenience I will also use the notation P_{id}^t to refer to the probability of an extant ancestral residue being followed by a nonempty cumulative insertion and a nonempty cumulative deletion; the relation between P_{id}^t and A_{id}^t is then $A_{id}^t = P_{id}^t P_m^t$. The derivative of A_{id}^t is more complex than the others above and has smaller impact. There are six types of events contributing to it: insertions after match columns that already had nonempty cumulative deletions but empty cumulative insertions (see e.g., Fig. 3f), at rate $(A_d^t - A_{id}^t)r_i$; deletions removing match columns followed by nonempty cumulative insertion and deletion, decreasing A_{id}^t at rate $A_{id}^t r_d/(1-g_d)$; deletions that introduce nonempty cumulative deletions at match columns that already have nonempty cumulative insertions but empty cumulative deletions (see e.g., Fig. 3G), at rate $(A_i^t - A_{id}^t)r_d/(1-g_i^t g_d)$ (the denominator considers that such deletions can start both within or at the end of the considered cumulative insertion, and uses the assumption that cumulative insertion length is independent of cumulative deletion length in the same chop zone); deletions that cause match columns (previously with empty cumulative insertions) to gain nonempty cumulative insertions by deleting intermediate match columns (see e.g., Fig. 3h), at rate $(P_m^t - A_i^t)r_d P_i^t(1-g_d)/((1-g_d(1-P_i^t))(1-g_d g_i^t))$; deletions that remove whole cumulative insertions from match columns previously followed by both nonempty cumulative insertion and deletion (see e.g., Fig. 3I), decreasing A_{id}^t at rate $A_{id}^t r_d(1-g_i^t)(1-g_d)/((1-g_i^t g_d)(1-(1-P_i^t)g_d))$; and lastly, deletions that remove an entire cumulative insertion after an unaffected match column that was previously followed by an empty cumulative deletion, and at the same time deleting a match column followed by a non-empty cumulative insertion, and not deleting this whole cumulative insertion (see e.g., Fig. 3j), with rate $(A_i^t - A_{id}^t)r_d P_i^t(1-g_d)g_d(1-g_i^t)/((1-(1-P_i^t)g_d)(1-g_d g_i^t)(1-g_i^t g_d))$. Combining all six terms, one obtains:

$$\begin{aligned}\frac{dA_{id}^t}{dt} &= (A_d^t - A_{id}^t)r_i - A_{id}^t r_d/(1-g_d) \\ &+ (A_i^t - A_{id}^t)r_d/(1-g_i^t g_d) \\ &+ (P_m^t - A_i^t)r_d \frac{P_i^t(1-g_d)}{(1-(1-P_i^t)g_d)(1-g_d g_i^t)} + \end{aligned}$$

$$-A_{id}^t r_d \frac{(1-g_i^t)(1-g_d)}{(1-g_i^t g_d)(1-(1-P_i^t)g_d)}$$
$$+(A_i^t - A_{id}^t) r_d \frac{P_i^t(1-g_d)}{(1-(1-P_i^t)g_d)(1-g_d g_i^t)} \frac{g_d(1-g_i^t)}{1-g_i^t g_d}. \quad (10)$$

The three derivatives in Equations 4, 6, 8, and 10 form a system of ordinary differential equations for the variables A_i^t, A_d^t, and A_{id}^t with initial point $t=0$ that I solve numerically using the odeint function of the scipy Python package (Bressert 2012). This typically requires a running time in the order of 10^{-3} s. Solving this system allows calculation of values for A_i^t, A_d^t, and A_{id}^t for any divergence time t and any instantaneous parameters r_i, r_d, g_i and g_d. From these (and considering equations above including 7 and 9) one can obtain values for g_i^t, g_d^t, P_i^t, P_d^t, and P_{id}^t. Finding the value of these variables is a prerequisite for defining and using the cumulative indel pairHMM described in the following section, and therefore for performing efficient alignment and parameter inference under the cumulative indel model. The differential equations above need to be solved for each combination of parameter values t, r_i, r_d, g_i, and g_d considered; they do not depend on the particular sequences to be aligned.

The Cumulative Indel pairHMM

The results from the previous section allow the definition of the cumulative indel pairHMM. Assuming that one sequence is descended from the other with divergence time t, a classical finite pairHMM has three states: M (for match, homology), D (for deletion), and I (for insertion). In Supplementary Section S6 available on Dryad, I discuss the more typical case of two sequences descended from a common ancestor. While in reality the case that one sequence is ancestor of the other is rarely met, it is however almost never possible to assign a directionality to indel events (and most substitution events) using just two sequences. As such, while the model considered is not necessarily stationary or reversible, I always make inference under the more computationally efficient assumption that one sequence is ancestor of the other. I instead simulate sequences under the GIM assuming they both descended from a common ancestor, so to account for the effects of likely model misspecification affecting the accuracy of the cumulative indel model. While not assuming stationarity means that the position of the root can affect inference, it also means that the model is better equipped to describe scenarios where sequence evolution is not at equilibrium.

Each state of the pairHMM refers to a type of pairwise alignment column: columns with M have residues in both sequences, columns with D only in the ancestral sequence, and columns with I only in the descendant. I define the transitions between states of the pairHMM in such a way that the probability of an alignment under such pairHMM corresponds to the probability of the sequence of cumulative indels in the model described above. Here, I assume that the considered alignment has been sampled by selecting a finite contiguous stretch of columns from an infinite alignment. This is often a reasonable approximation since sequences within an alignment are usually selected from within much longer chromosomes. I assume that all possible lengths of the sampled alignment are equally likely, which corresponds to a uniform prior measure over alignment lengths. This means that the probabilities of all possible alignments of all possible sequences will not sum up to one, but the probabilities of all alignments of the same length will. Note that a uniform prior over alignment lengths does not necessarily yield a uniform prior over the length of either the ancestral or the descendant sequence length.

Given the assumption that the considered alignment starts at a random position within an infinite alignment, the probabilities of a state at the first alignment column is defined as its equilibrium probabilities within the pairHMM. Equivalently, if one denotes as S the start state of the pairHMM, we have the following transition probabilities:

$$P(X|S) = P(X), \text{ for } X = M, D, I, \quad (11)$$

where equilibrium probabilities $P(M)$, $P(D)$, and $P(I)$ can be calculated from the transition probabilities defined below. A pairHMM describes the probability that one type of column is followed by another type of column, assuming the Markov property along the pairwise alignment. The transition probabilities $P(S_2|S_1)$ from state S_1 to state S_2 in the cumulative indel pairHMM are defined as:

- $P(M|M) = 1 - P_i^t - P_d^t + P_{id}^t$
- $P(D|M) = P_d^t$
- $P(I|M) = P_i^t - P_{id}^t$
- $P(M|D) = (1-g_d^t)(P_d^t - P_{id}^t)/P_d^t$
- $P(D|D) = g_d^t$
- $P(I|D) = (1-g_d^t)P_{id}^t/P_d^t$
- $P(M|I) = 1 - g_i^t$
- $P(I|I) = g_i^t$.

Lastly, we ignore the contribution of the probability of terminating the alignment from any column, that is, of transitioning from any state M, D, I to the end state E representing the end of the alignment. This is equivalent to considering the length of the alignment known a priori, that is, conditioning on alignment length. This in practice does not affect the algorithm

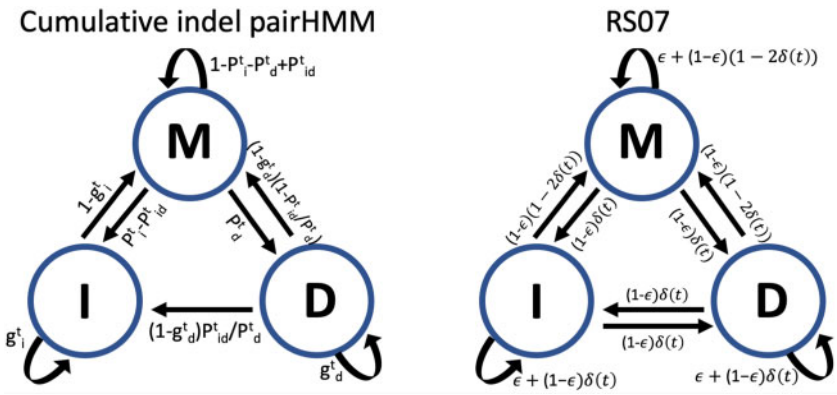

FIGURE 4. PairHMM structure. Graph representation of the cumulative indel pairHMM (left) compared to the RS07 (Redelings and Suchard 2007) pairHMM (right). Hidden states are represented as a graph vertices (circles) and non-zero transition probabilities as edges (arrows). The RS07 model has been here simplified by removing start and end states. Unlike in the RS07 model, in the cumulative indel model the transition from deletion (D) to insertion (I) column is never allowed.

implementation, but has the consequence that the alignment probabilities (of alignments of any length) do not sum to 1. As mentioned above, this is equivalent to defining a measure over alignments rather than a probability, which has the benefit of allowing uniform prior weights over possible alignment lengths. More detail for the rationale behind the definition above of transition probabilities and their relation to the cumulative indel probabilities is given in Supplementary Section S2 available on Dryad.

The assumption of how an alignment is sampled from an infinite alignment might not be realistic in certain common situations. In the Supplementary Section S4 available on Dryad, I define more realistic alignment boundary conditions at the expense of increased model complexity (but small additional computational complexity). The key modifications necessary for these extensions is to alter the alignment starting and ending probabilities $P(X|S)$ and $P(E|X)$ defined above. In Supplementary Section S5 available on Dryad, I also discuss how to define a cumulative indel pairHMM for the probability of a descendant sequence and alignment, conditional on the ancestral sequence and parameters, as is typical for pairHMMs used in multiple sequence alignment and phylogenetic inference.

Figure 4 presents a graphical representation of the cumulative indel pairHMM. There are some noticeable differences between the cumulative indel pairHMM and previous finite pairHMMs (i.e., previous pairHMMs except the long indel model). An important difference is that the transition probability from I to D is 0 as cumulative insertions are required to appear in the alignment after deletions; this has the effect of reducing the number of alignments with nonzero probability, in turn simplifying the search of an optimal alignment, and integration over alignment space. In fact, in the cumulative indel pairHMM, a chop zone is represented by a single possible chop zone alignment (all deletion columns followed by all insertion columns). This is very different from the TKF91 and TKF92 pairHMMs where many alignments can be possible for the same chop zone, and different alignment represent different evolutionary histories leading to the same chop zone. In the cumulative indel pairHMM, all possible evolutionary histories are still considered, but now allowed alignments typically have more evolutionary histories consistent with them. Assuming 0 transition probability from I to D also reduces the theoretical accuracy achievable by a general pairHMM, that is, it could be possible, by removing the constraint $P(D|I) = 0$, and by adjusting the other transition probabilities accordingly, to define a more accurate pairHMM than the cumulative indel pairHMM. A specific example of this is when simulating indel events of length 1 (i.e., simulating under the assumptions of the TKF91 model). In this case, the TKF91 pairHMM matches the simulated patterns exactly, while the cumulative indel pairHMM introduces an element of approximation (see next section).

Here, I do not assume the existence of sequence fragments, unlike most previous pairHMMs. A graphical comparison of the cumulative indel pairHMM and the RS07 pairHMM (Redelings and Suchard, 2007) is given in Figure 4 while a comparison to other finite pairHMMs is presented in Supplementary Section S16 available on Dryad.

Testing the Cumulative Indel pairHMM with Simulations

I use INDELible (Fletcher and Yang 2009) to simulate sequence evolution under the GIM and to test how well the pairHMM above and other pairHMMs fit the dynamics of the GIM (Figure 5 and Supplementary Figs. S2–S4 available on Dryad). I simulate an ancestral 2 Mbp sequence and alignments at different levels of divergence, from which I extract the proportions of nonempty cumulative indels and the distributions of their lengths. I then compare these values to those predicted by the cumulative indel model, by the TKF91, TKF92, RS07, and PRANK (Löytynoja and Goldman 2005, 2008a, 2010) finite pairHMMs, and by the long

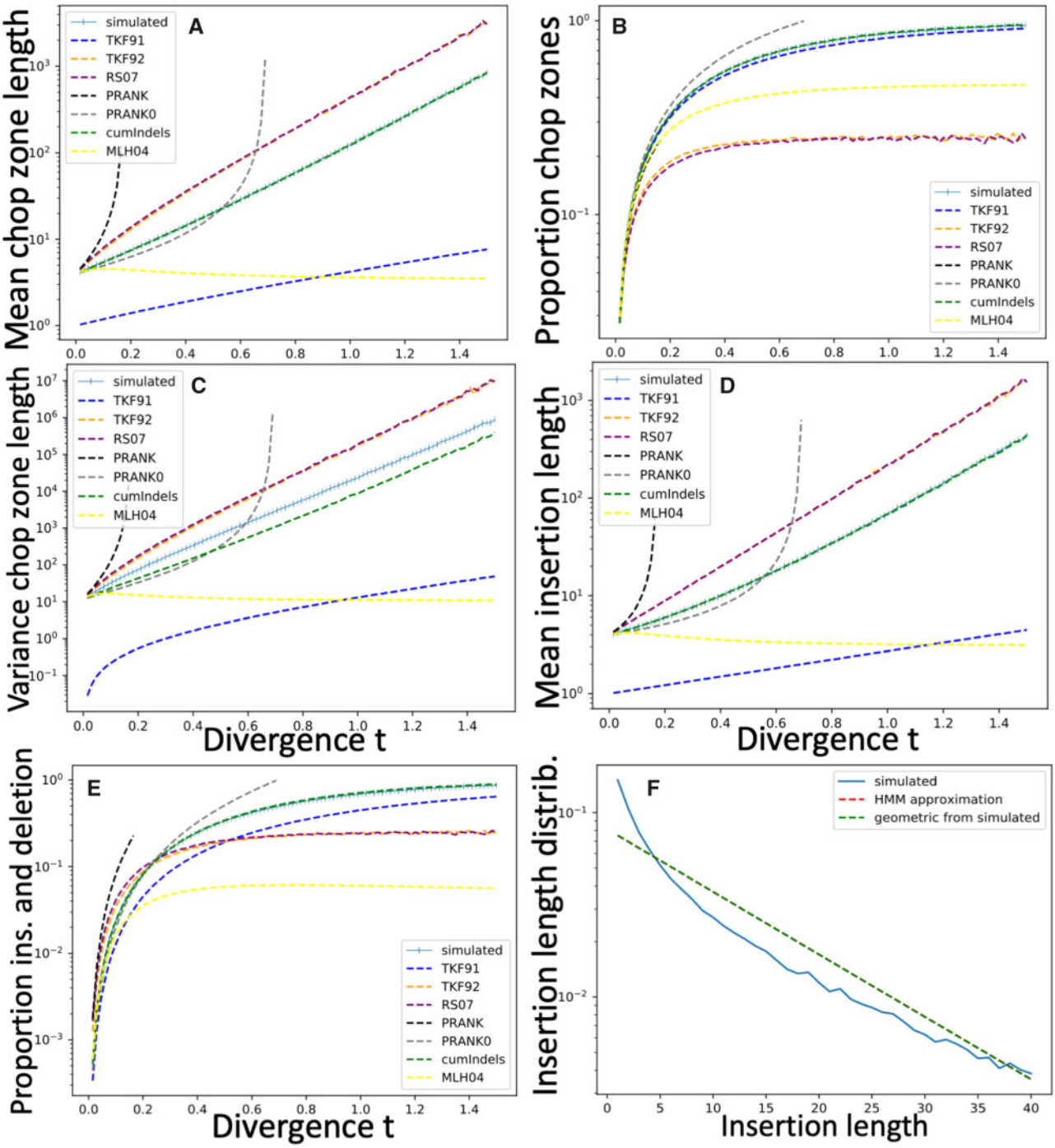

FIGURE 5. Comparison of pairHMM predictions. INDELible simulations are compared with patterns predicted by different pairHMMs. Simulated indel rates are $r_i = r_d = 1.0$, and simulated instantaneous indel lengths are geometrically distributed with mean 4 ($g_i = g_d = 0.75$). a–e) For each of the 100 values of t on the X axis (ranging from 0.015 to 1.50), I simulated evolution of an ancestral genome of 2 Mbp. Azure lines show simulated values. Green dashed lines, often overlapping with simulated values, show the predictions from the cumulative indel pairHMM. The other colors show the predictions of TKF91, TKF92, PRANK, RS07, and MLH04 models. a) Mean length of non-empty chop zones. b) Proportion of non-empty chop zones. c) Variance in non-empty chop zone length. d) Mean length of non-empty cumulative insertions. e) Proportion of chop zones with both non-empty cumulative deletions and non-empty cumulative insertions (P^t_{id}). f) Distribution of non-empty cumulative insertion lengths simulated at divergence $t = 0.5$ (azure line) versus expected from the cumulative indel model (red line) and a geometric distribution fit to the observed distribution (green line, overlapping with the red line). In all models, indel rates are set so that the expected number of indels per unit time at $t \approx 0$ are the same as in simulations (for the TKF91 and TKF92 I set equal insertion and deletion rates; see Rivas and Eddy 2015). For TKF92, PRANK and RS07, I set indel fragment lengths equal to the simulated mean instantaneous indel length. In PRANK, I either set the match block extension probability $\gamma = 0$ (gray line) or equal to the indel block extension probability $\gamma = \epsilon$ (black line). In MLH04, I set as approximations the maximum number of indel events per chop zone to 3, and the maximum chop zone length to 35.

indel model generalized pairHMM (MLH04, Miklós et al. 2004). For the cumulative indel model and the MLH04 I use the true (simulated) instantaneous indel rates, divergence time, and instantaneous gap extension probabilities as parameter values. To allow calculation of MLH04 expected features, I assume, as an approximation, a maximum of 3 indel events per chop zone and maximum chop zone length of 35, and I use the implementation of the MLH04 in https://github.com/ihh/trajectory-likelihood/. The TKF91, TKF92, RS07, and PRANK pairHMMs have different parameterizations than the simulated model (e.g., assuming unsplittable fragments or 1-residue instantaneous indels), so they are not straightforward to compare to the cumulative indel model (see e.g., Fig. 4 and Supplementary Fig. S16 available on Dryad) and to simulations. Therefore, in the TKF91, TKF92, RS07, and PRANK pairHMMs, I set, as an example, parameter values that fit well simulations at short divergence time ($t \approx 0$) in terms of numbers of expected indel events per time unit; if possible (i.e., for non-TKF91 models) I also set parameter values to match simulated indel lengths at $t \approx 0$ (so gap extension probabilities equal to simulated g_i and g_d). For PRANK, I set the match fragment extension probability γ to either $\gamma = 0$ or $\gamma = \epsilon$. For a more detailed description of all finite pairHMMs considered here, see Supplementary Section S9 available on Dryad. I extract predicted patterns of cumulative indel proportions and lengths distributions in finite pairHMMs from alignments of 2 million columns simulated under the parameter settings of each finite pairHMM. For the MLH04 instead, I calculate the probabilities of different chop zones under the given parameter values and the given approximations, normalize them, and use them to calculate the considered expected chop zone patterns.

The cumulative indel pairHMM closely tracks many simulated patterns, including mean lengths of chop zones, cumulative insertions, and cumulative deletions, and proportions of non-empty chop zones and non-empty insertions (Fig. 5 and Supplementary Fig. S2 available on Dryad). Other patterns are not perfectly matched, but seem more closely matched compared to other pairHMMs, for example median cumulative indel lengths, proportion of non-empty cumulative deletions, and variance of cumulative indel lengths. Most predictions by the TKF91 model are strongly affected by its assumption of 1-residue instantaneous indels. The PRANK pairHMM seems instead mostly affected by the fact that its transition probabilities are only defined up to a threshold of divergence, and that, approaching this threshold, expected cumulative indel lengths diverge to infinity. The TKF92 and RS07 models are affected by their fragment assumption, in that, most remarkably, the expected proportion of non-empty chop zones converges to a relatively low probability instead of approaching 1.0 as divergence increases. The MLH04 model, while in theory closely matching the simulated model, in practice seems affected by the approximations (maximum number of indels in chop zone and maximum chop zone length) used to implement the model in practice; in particular, at high divergence, the model seems to considerably underestimate cumulative indel lengths. This could be addressed by relaxing the approximations further; however, this would also come at considerable computational cost.

The cumulative indel pairHMM assumes geometrically distributed cumulative indel lengths for any t. Even if this assumption can be met at $t \approx 0$, simulated cumulative indels at high t are too dispersed to be fit by a geometric distribution (Fig. 5F, Supplementary Fig. S3 available on Dryad and Rivas and Eddy, 2015) and as such cannot be accurately described by a pairHMM.

While, in the simulations above, the cumulative indel pairHMM seems to represent a better approximation to the GIM than other pairHMM, this is not necessarily the case for all theoretical scenarios. For example, when considering indel events that only affect 1 residue at the time, that is, under the assumptions of the TKF91 model, TKF91 predictions are exact, while the cumulative indel model ones are only approximations (see Supplementary Fig. S4 available on Dryad). In this scenario, the predictions of the cumulative indel pairHMM are very close to the simulations, but the expected variance of non-empty chop zone lengths does not perfectly match them. One of the reasons for this is that the cumulative indel pairHMM has the constraint $P(D|I) = 0$, and this does not allow it to accurately describe the correlation between cumulative indel length and cumulative deletion length within the same chop zone. Relaxing the constraint $P(D|I) = 0$ (and consequently adjusting the other transition probabilities) might therefore lead to a slightly more accurate pairHMM, which would however still be an approximation in most realistic scenarios due to the non-Markov nature of alignments generated under the GIM.

Substitution Probabilities

So far I discussed patterns of residue presence–absence, that is, the indel process, and ignored substitutions and sequence composition. As usual in statistical alignment (Mitchison and Durbin 1995; Fleissner et al. 2005; Redelings and Suchard 2005; Fletcher and Yang 2009), I assume that the substitution probabilities are independent of the indel process. In Supplementary Section S7 available on Dryad, I discuss how this independence follows from typical assumptions regarding substitutions (substitution process at equilibrium) and indels (indel rates independent of sequence composition, and inserted sequence composition sampled from equilibrium residue frequencies) which I also assume hereby.

The phylogenetic probability P_{sub} of a single alignment column, assuming that one sequence is the ancestor of the other with divergence time t, is:

- $P_{\text{sub}}((R_1,-))=P((-,R_1))=\pi(R_1)$
- $P_{\text{sub}}((R_1,R_2))=\pi(R_1)S^t_{R_1R_2}$

where R_1 and R_2 are residues, $(R_1,-)$ and $(-,R_1)$ are respectively a deletion and an insertion alignment column, (R_1,R_2) is a match alignment column, π are the equilibrium residue frequencies, and $S^t_{R_1R_2}$ is the probability of having residue R_2 in the descendant sequence conditional on its homologous ancestor being R_1. Probability $S^t_{R_1R_2}$ is entry (R_1,R_2) of the probability matrix $S^t=e^{tQ}$, where Q is the instantaneous residue substitution rate matrix (Yang and Rannala 2012). In the rest of the manuscript, I consider the case of amino acid residues, and I assume that substitution rate Q and amino acid frequencies π are from the LG model (Le and Gascuel 2008). For nucleotide sequences, entirely analogous methods can be employed.

In some cases, one is interested in the probability of the descendant sequence conditional on the ancestral sequence (e.g., in treeHMMs like BAli-Phy, Redelings and Suchard 2005). In this case, given that the first sequence is ancestral (e.g., in match column (R_1,R_2) the ancestral residue is R_1) one has:

- $P_{\text{sub}}((R_1,-))=1$
- $P_{\text{sub}}((-,R_2))=\pi(R_2)$
- $P_{\text{sub}}((R_1,R_2))=S^t_{R_1R_2}$

Dynamic Programming Algorithms

Just like any other classical finite pairHMM, one can use the cumulative indel model within classical finite pairHMM algorithms for statistical inference of pairwise alignments (Needleman–Wunsch algorithm), evolutionary parameter inference (Baum-Welch algorithm), indel history inference (Viterbi algorithm), and posterior decoding (Forward and Backward algorithms, Durbin et al. 1998; Lunter 2007a).

Given two sequences s_1 and s_2 of length l_1 and l_2, all these methods employ a dynamic programming matrix L with l_1+1 rows and l_2+1 columns. Entry $L_{j,m}$ of L, with $0\le j\le l_1$ and $0\le m\le l_2$, refers to probabilities of partial alignments of the first j residues of sequence s_1, $(s_1^1,\ldots,s_1^j)$, and the first m residues of s_2, $(s_2^1,\ldots,s_2^m)$. For example, in the pairHMM version of the Needleman–Wunsch algorithm, $L_{j,m}$ tracks information regarding the highest partial alignment probability of $s_1^1,\ldots,s_1^j$ and $s_2^1,\ldots,s_2^m$. Entries $L_{j,m}$ are calculated dynamically, starting from $L_{0,0}=1$, and ending with $j=l_1,m=l_2$. In order to calculate $L_{j,m}$, only the values of $L_{j-1,m}$, $L_{j,m-1}$, and $L_{j-1,m-1}$ are usually needed. The main difference between this method and classical pairwise aligners is that one uses pairHMM state transition probabilities (like the ones in Section "The cumulative indel pairHMM") instead of fixed and arbitrary gap opening and extension penalties, and phylogenetic substitution probabilities (like the ones in Section "Substitution probabilities") instead of fixed mismatch penalties. In case of the pairHMM Needleman–Wunsch, the final dynamic programming score now has a clear interpretation as the highest alignment log-probability between the two considered sequences and given a set of evolutionary parameter values. The algorithm also allows the reconstruction of the specific alignment with the highest probability. I do not employ evolutionary parameter values fixed a priori, but I estimate them from the sequences themselves using similar dynamic programming methods. In Supplementary Section S8 available on Dryad, I describe the cumulative indel model versions of these methods and some extensions.

All methods were implemented in Python and are available from https://bitbucket.org/nicofmay/cumulativeindel. I implemented pairHMMs for the cumulative indel model, as well as for the TKF91, TKF92, RS07, and PRANK models. While the values of state transition probabilities differ among models, as do their parameterizations, most of the pairHMM techniques discussed here and in Supplementary Section S8 available on Dryad remain the same for different pairHMMs. For efficiency, I represent and calculate probabilities as "more buoyant floats" (Lunter 2007a) which is a particularly convenient number representation for pairHMMs. To further reduce computational demand, I often run Python scripts with the PyPy2 v6.0.0 (https://pypy.org/) alternative Python implementation.

Fast Dynamic Programming Approximation: Adaptive Banding

All PairHMM dynamic programming (DP) algorithms described in Section "Dynamic programming algorithms" and Supplementary Section S8 available on Dryad work on a DP matrix L of size $(l_1+1)\times(l_2+1)$. When the sequences considered, s_1 and s_2, are close relatives, most of these partial alignments considered in L are very unlikely. This has sparked interest in "banding" or "corner cutting" approaches (Chao et al. 1992; Hein et al. 2000; Havgaard et al. 2007; Westesson et al. 2012; Bogusz and Whelan 2017) that aim to focus only on likely regions of L.

Here, I propose and adopt a new such approach called "adaptive banding" (see Fig. 6). Adaptive banding fills L along diagonals, unlike typical column- or row-wise approaches. Each adaptive banding iteration i with $0\le i\le l_1+l_2$ considers cells (j,k) with $j+k=i$, so that each partial alignment in the current iteration has a constant number of residues, and therefore comparable data size and probability (Fig. 6b,d). If a cell has much lower

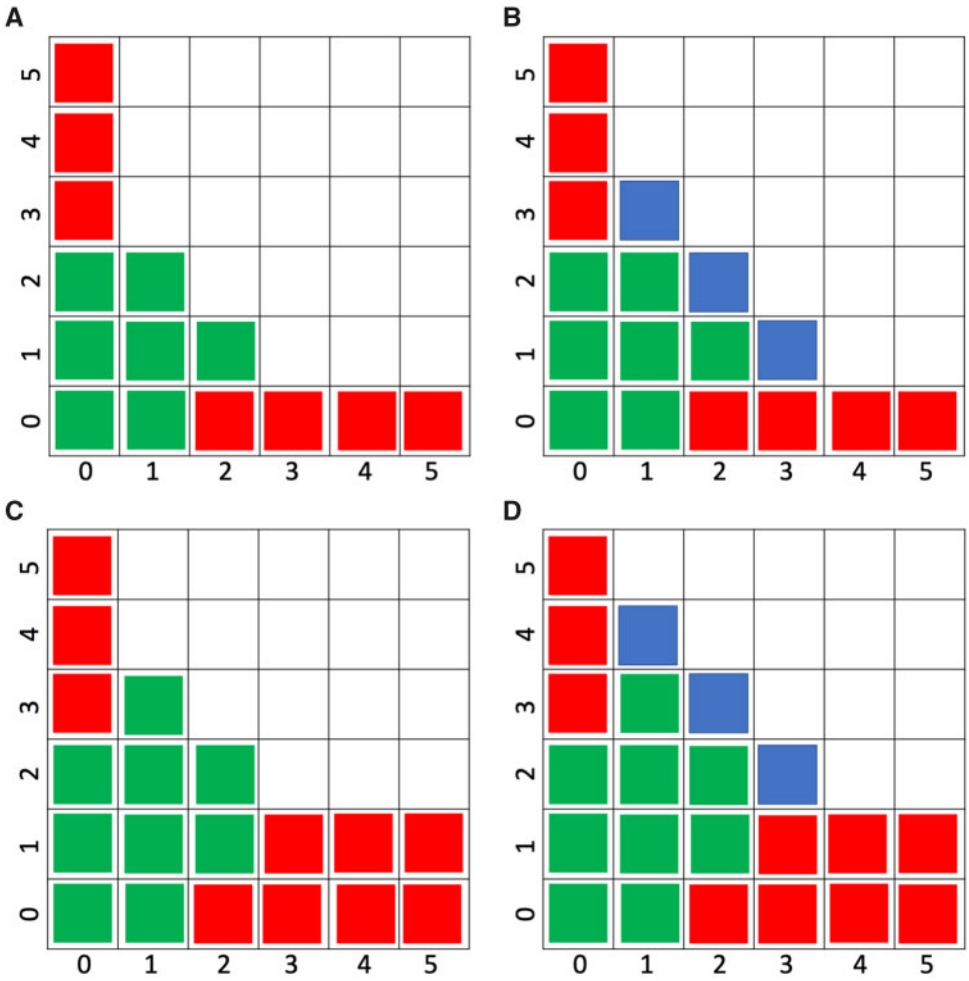

FIGURE 6. Graphical example of adaptive banding. Two sequences of length 5 are considered. Green cells have been previously considered and not discarded, red cells have been discarded, and blue ones are being considered in the current iteration. From a) to b), a new iteration evaluates cells along a diagonal, but cells in the bottom row and leftmost column were already discarded and are ignored. c) Cell (1,3) is discarded in the current iteration, and therefore so are the other (not yet evaluated) cells in row 1 and d) a new iteration starts. At each iteration, any number of rows and columns can be discarded.

probability (or score) than some other cells considered in the same iteration, it is discarded for the rest of the algorithm. In all applications below, unless otherwise mentioned, the minimum threshold for discarding cells is set to 15 log-probability units difference. Further, if a bottom cell (i.e., one with only discarded cells under it in the same column) is discarded, all the cells on its right in the same row can also be discarded, as they become unreachable except through previously discarded cells (Fig. 6c). Similarly, if a left-most cell is discarded, all cells above it in the same column are also discarded.

In the ideal scenario that two sequences are homologous and with high enough identity, adaptive banding can considerably reduce the time cost of all dynamic programming methods by filling only a narrow band of L surrounding the highest-probability alignment. However, in the worst-case scenario of two non-homologous (or extremely diverged) sequences, because all cells in the same adaptive banding diagonal are expected to have similar probabilities, the cost of this method can remain proportional to the size of L.

To reduce not only time demand, but also memory demand of this approach, one can avoid to allocate memory for DP matrix cells that are discarded without being visited. I achieve this by allocating memory for one diagonal at the time just before said diagonal is filled. Instead of allocating memory for all cells in a diagonal, I only allocate memory for the cells in the diagonal that have not been already discarded. Then, to be able to correctly access a cell, I only need to record how many cells have not been allocated from the bottom of each diagonal. In a favorable scenario with two similar homologous sequences, this approach leads to a significant saving in memory, and to comparable memory and time demands.

Many previous banding approaches have assumed a fixed prior band size, considering only cells of the

dynamic programming matrix that fall within this predetermined diagonal band (see Chao et al. 1992 and citations within). Adaptive banding has the advantage that band size is not determined a priori, allowing long indels within an alignment, but also allowing a reduced band size in regions of high similarity where alignment uncertainty is low. (Hein et al., 2000) proposed instead to use a similarity alignment between two sequences to delimit a band around the maximal similarity alignment, the size of the band being determined by the similarity scores of the suboptimal alignments. Adaptive banding has the advantage with this respect of not requiring an initial similarity alignment, therefore allowing the alignment of very long but similar sequences; also, adaptive banding bases the size and location of the band on the model used for statistical alignment, which might prevent biases in the case the similarity alignment model would support different alignments than the statistical alignment model. (Havgaard et al., 2007) proposed a banding (or "pruning") scheme in which a threshold for a minimum acceptable subalignment score is defined based on the length of the aligned subsequences. While this approach shares several aspects with adaptive banding, it has the limitation that when the aligned sequences are not very similar, or if there are long indels, sometimes no acceptable alignments can be found at all. Adaptive banding has instead the advantage that the subalignment scores are compared with each other, preventing this problem and allowing many subalignments to be considered in regions of high alignment uncertainty. (Westesson et al., 2012) implemented a banding approach within an MCMC framework. In their method, a new sampled alignment has to reside within a band of fixed size around the currently considered alignment. Adaptive banding could be useful in this context to automatically modulate band size relative to alignment uncertainty (e.g., a larger band for long tree branches and a small band for closely related sequences) and relative to alignment uncertainty locally along the alignment. (Bogusz and Whelan, 2017) utilized a more complex banding scheme with an initial banding based on k-mer distance estimates, followed by identification of high posterior cell using the Forward–Backward paradigm. Adaptive banding does not need initial k-mer based assessment, and as such might work even in cases where divergence might disrupt most k-mers while still leaving a detectable signal of homology. Adaptive banding was developed independently of another similarly named approach (adaptive banded dynamic programming, Suzuki and Kasahara, 2017) where the size of the band is fixed before running dynamic programming alignment, while the location of the band within the dynamic programming matrix is allowed to change.

Adaptive banding can be used not only with the cumulative indel model pairHMM methods, but also with other pairHMMs (such as the TKF91, TKF92, RS07, and PRANK models), as well as with classical dynamic programming alignment methods. Here, I implement and use adaptive banding for all finite pairHMM models considered in this text.

Simulations

I use simulations in INDELible (Fletcher and Yang 2009) with geometric instantaneous indel length distributions to compare different methods of alignment and parameter inference. I will instead not use benchmark structural alignments, since the different assumptions of structural alignments (used in these data sets) and evolutionary alignment (considered in the current work) can cause considerable biases (Iantorno et al. 2014; Tan et al. 2015). I simulate substitutions under an LG model (Le and Gascuel 2008) and assume that the amino acid rate matrix and frequencies are known during inference. Simulated instantaneous indel gap extension probabilities are always $g_i = g_d = 0.75$ (so instantaneous indels have an average length of four residues).

All considered finite pairHMM methods (the cumulative indel pairHMM, but also the TKF91, TKF92, RS07, and PRANK ones) were run using adaptive banding (unless stated otherwise) and with custom Python scripts available from https://bitbucket.org/nicofmay/cumulativeindel/.
In each scenario and data set, I perform dynamic programming pairHMM alignment inference with a set of fixed parameter values specific for the considered data set. These parameter values are estimated via maximum likelihood from the simulated data itself, and as such they are expected to fit well each simulated data set (see e.g., Lunter 2007b for a similar approach), and therefore to be a sensible choice for performing alignment. Maximum likelihood parameter inference is performed with a dynamic programming pairHMM method, the Forward algorithm (Supplementary Section S8 available on Dryad), which calculates the likelihood of pairHMM parameter values given two homologous sequences by efficiently integrating over all their possible alignments. I estimate maximum likelihood parameter values for a dataset by using the pairHMM Forward algorithm over different parameter values combinations, until a local likelihood maximum in parameter space is reached; specifically, I use the Nelder–Mead method in the scipy.optimize package (Gao and Han 2012) to explore parameter space and optimize the likelihood function given by the pairHMM Forward algorithm. For each maximization, I start from three different points in parameter space ($(t = 0.2, r_i = r_d = 0.1, g_i = g_d = 0.5)$, $(t = 0.1, r_i = r_d = 0.4, g_i = g_d = 0.5)$, and $(t = 0.1, r_i = r_d = 0.1, g_i = g_d = 0.9)$) to help in case non-global likelihood maxima are present. To aid parameter inference, and to address the fact that in pairwise alignment normally insertions and deletions are not distinguishable from one another (unless indeed one sequence is ancestor of the other), I fix $r_i = r_d$ and $g_i = g_d$ in the cumulative indel model; similarly, I fix $\lambda = \mu$ in TKF91 and TKF92 (where λ is the instantaneous

insertion rate and μ the instantaneous deletion rate), and either $\epsilon = \gamma$ in the PRANK model (where ϵ is the gap fragment extension probability and γ the match fragment extension probability) or $\gamma = 0$. I do not condition on ancestral sequence length for any pairHMM, and I do not assume that ancestral sequence length is at equilibrium. For each data set, I then use the inferred maximum likelihood parameter values as fixed parameters for alignment inference with the pairHMM Needleman–Wunsch algorithm (see Supplementary Section S8 available on Dryad). For comparison, I also performed alignments with the traditional pairwise alignment implementation of the NEEDLE function in EMBOSS v6.6.0 (Rice et al., 2000) with default options; this latter method is a traditional aligner without an explicit probabilistic model, but with gap opening, gap extension, and mismatch penalties. I consider these penalties as fixed parameters, so they do not change across applications to simulated data sets.

To investigate computational demands, I use pairwise alignments of different lengths and simulated under different evolutionary parameters (see Supplementary Fig. S5 available on Dryad).

To test parameter and alignment inference, I consider instead two simulation scenarios, each with 150 simulated pairs of homologous sequences evolved from a 1 kb common ancestor. While in pairHMM inference, I assume that one sequence in each pair is ancestral to the other (so as to reduce computational cost), in simulations both sequences descend from a common ancestor with equal divergence times (so as to increase realism of simulations). This represents a model misspecification for the cumulative indel model as it does not assume stationarity. In the first simulation scenario ("1 species pair"), I assume that 150 homologous gene pairs are selected always from the same two species; the divergence time t is then inferred together with indel rate $r = r_i = r_d$ and instantaneous gap extension probability $g = g_i = g_d$. This scenario addresses the accuracy of inference when it is known that all pairwise alignments have similar levels of divergence. In the second simulation scenario ("150 species pairs"), I assume that all 150 homologous gene pairs are selected from different species pairs; the divergence time t for each gene pair is sampled from a uniform distribution over $[0, 1]$; during inference, divergence times are assumed known and only r and g are inferred. In this scenario, as in treeHMMs and in (Bogusz and Whelan, 2017), the same pairHMM parameters have to fit different levels of divergence simultaneously. Assuming known divergence times might not be generally realistic, however, in practice, inferring 150 divergence times and indel parameters simultaneously with the methods considered here would be excessively computationally demanding; instead, in scenarios when divergence time is unknown it would be feasible to first infer a guide tree, as done by most multiple sequence alignment methods, and then use the guide tree to inform divergence times for indel parameter inference, similar to what is presented here.

Results

Computational Demand

Both classical alignment and finite pairHMM algorithms are expected to have quadratic time costs in sequence length. Simulations confirm this pattern (Fig. 7 and Supplementary Fig. S5 available on Dryad); however, adaptive banding makes alignment much faster, approximately linear in sequence length. Long sequence statistical alignment with adaptive banding is faster than with EMBOSS, and adaptive banding even allows genome-wide statistical alignment (see inference from human-chimp synteny block below). Despite this, short sequence alignment with EMBOSS remains faster than with pairHMM alignment. This is expected due to the greater computational complexity of operations on probabilities than of operations on integer scores. However, a major part of the computational demand difference is due to the specific Python implementation considered here of the pairHMM methods. This interpreted high-level language introduces slow-downs, in particular when compared with the optimized C implementation of EMBOSS, and could be addressed by re-implementing all methods in C. Solving the differential equations of the cumulative indel model requires negligible time (in the order of 10^{-3} s per alignment) and so all the finite pairHMMs considered here have similar computational demands.

The computational demand of adaptive banding is very dependent on the level of uncertainty in the

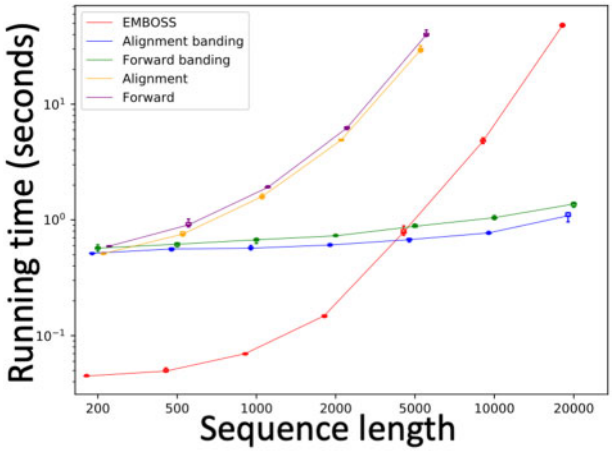

FIGURE 7. Comparison of running times. Computational demand of classical and pairHMM alignment methods. On the Y axis are running times in seconds; on the X axis the lengths of simulated ancestral sequences. Here, simulated parameters are $t = 0.05$, $r_i = r_d = 0.05$, and $g_i = g_d = 0.75$. Each boxplot includes 10 replicates, and the bars within boxplots represent median, lower and upper quartiles, and extreme values. Boxplot bars are often not clearly visible due to differences in running time within methods being much smaller than differences between methods. "Alignment banding" and "Alignment" refer to the cumulative indel pairHMM Needleman–Wunsch algorithm, with and without banding respectively, and with simulated parameter values. "Forward banding" and "Forward" refer to the cumulative indel pairHMM Forward algorithm, with and without banding, respectively. Values for non-banding pairHMMs for 10 kb and 20 kb sequences were not calculated due to elevated computational demand.

alignment. In fact, the width of the bands visited during alignment typically increases at high divergence and increase locally in the presence of long indels (Fig. 8). Therefore, adaptive banding is more computationally demanding for highly diverged sequences than for closely related ones. Furthermore, while an adaptive banding threshold of 15 log-likelihood units was sufficiently accurate for all scenarios simulated under the GIM, when hiding large parts of a sequence (or, equivalently, when introducing large artificial indels) adaptive banding could in certain cases miss the true alignment. In such cases, making the threshold stricter at 30 log-likelihood units was typically sufficient to address the issue (Fig. 8d–f and Supplementary Fig. S9 available on Dryad). In real life scenarios, however, when large portions of some sequences might be missing, or if very long indels could be present, care should be used in selecting a proper adaptive banding likelihood threshold; in cases where no prior upper limit for the size of missing parts of sequences is known, more nuanced banding strategies might be required (see "Discussion" section).

Accuracy of Parameter Inference

Parameter inference under the cumulative indel pairHMM appears accurate under all considered simulation scenarios (Fig. 9 and Supplementary Fig. S6 available on Dryad). The pairHMM parameter inference approach used here efficiently integrates over plausible pairwise alignments. An even more computationally efficient method, often used in practice, is to first infer a single alignment, and then infer parameter values from the fixed alignment. I recreated this second approach by first inferring alignments with EMBOSS and then performing cumulative indel pairHMM parameter inference on fixed alignments. This faster approach results in inaccurate estimates, as it is probably subject to the effects of alignment errors, in particular over-alignment, which could explain why t is overestimated and r underestimated (Fig. 9).

Among the finite pairHMMs, the TKF91 noticeably leads to the least reliable estimates, as inferred values of t and r are considerably different from those used for simulation. This is not surprising given that TKF91 only allows 1-residue instantaneous indels, and so needs multiple indel events to explain any long simulated indel event. Also noticeable is the underestimation of g by the TKF92 and RS07 pairHMMs (corresponding to ϵ in the original notation) at high divergence and their overestimation of r (Fig. 9). In these models, ϵ describes fragment lengths of both indel and match columns, and so the lack of simulated contiguous match columns at high divergence might cause its underestimation, and consequently also the overestimation of r. Similarly, the PRANK model with $\gamma=\epsilon$ tends to underestimate γ at high divergence, probably for similar reasons as the TKF92 and RS07 pairHMMs. When however one sets $\gamma=0$ in the PRANK pairHMM, this pattern changes considerably. Overall the cumulative indel pairHMM seems the most reliable at inferring evolutionary parameters, in particular the gap extension parameter g; the PRANK pairHMM with $\gamma=0$ also seems to perform similarly well. The fact that a model's parameter estimates do not match simulated values does not necessarily mean that the models is less accurate, but only that its parameter estimates are less clearly interpretable.

Accuracy of Alignment Inference

All finite pairHMMs considered here, except the TKF91 model, seem to lead to better alignments than the classical aligner EMBOSS in most simulation scenarios and parameter settings (Fig. 10 and Supplementary Fig. S7 available on Dryad). This is true in particular for the number of wrongly inferred match columns, suggesting that, especially at elevated divergence times, non-TKF91 pairHMMs reduce overalignment biases of traditional aligners. All non-TKF91 pairHMMs have similar performance across most data sets, in particular in the "1 species pair" scenario (Supplementary Fig. S7 available on Dryad). This suggest that all non-TKF91 pairHMMs are better equipped to model any individual level of divergence and indel rate, at least when paired with appropriate parameter inference procedures. The TKF91 pairHMM shows systematically the worst results (Fig. 10 and Supplementary Fig. S7 available on Dryad). This is not surprising when compared to other pairHMMs, considering that TKF91 assumes 1-residue instantaneous indels. However, TKF91 often also performs considerably worse than EMBOSS. This suggests that, while the TKF91 is an elegant model with interesting mathematical and computational properties, it might often be over-performed by simpler, score-based affine-gap models in practical applications.

In the "150 species pairs" simulation scenario at high indel rates, all non-TKF91 finite pairHMM also seem to perform similarly, and the cumulative indel model seems to show slightly fewer wrong match columns (Fig. 10), possibly because of its ability account for indel patterns expected at differing levels of divergence simultaneously. TKF92 seems to perform second best. The cumulative indel model and the PRANK model with $\gamma=0$ tend to show higher likelihoods (Supplementary Fig. S8 available on Dryad).

Human-Chimp Alignment and Parameter Inference

To showcase the applicability of the methods presented here, in particular with respect to parameter inference and alignment of long sequences, I analyzed a segment of shared synteny between human (reference hg38) and chimp (reference PanTro6). This was downloaded from the UCSC Genome Browser pairwise alignment of human and chimp https://hgdownload.cse.ucsc.edu/goldenPath/hg38/vsPanTro6/. Syntenic blocks and alignments in the

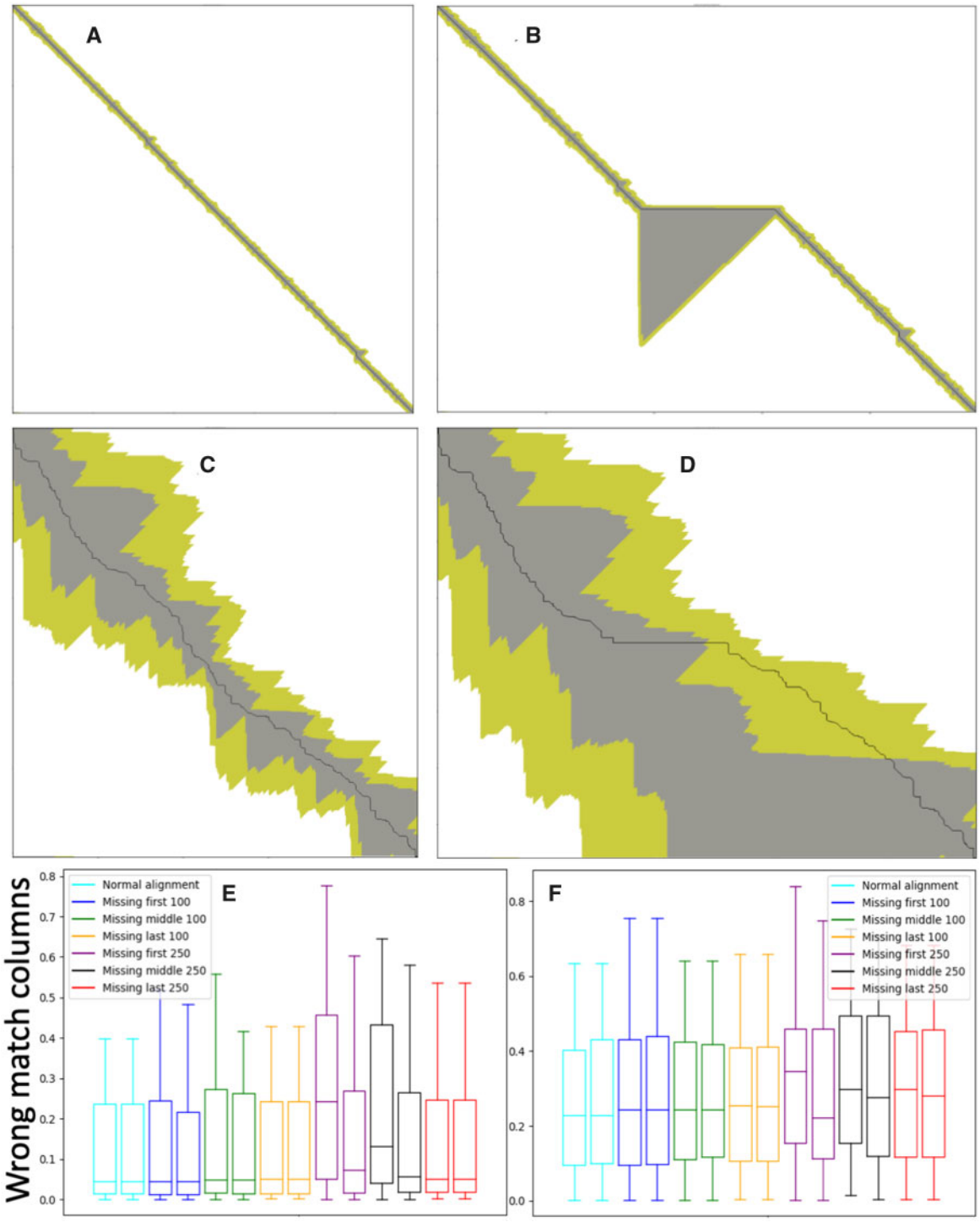

FIGURE 8. Adaptive banding performance and the effects of missing sequence data. a–d) Examples of dynamic programming matrices for alignments simulated under the "1 species pair" scenario. As typical for alignment dynamic programming matrices, each cell represents a partial alignment between subsequences of the two sequences being aligned. Here, the matrices are filled using adaptive banding, from the top left to the bottom right of each matrix. The black path represents the true simulated alignment between the considered sequences. The gray cells are those that are not discarded by adaptive banding with a log-likelihood threshold of 15 (the threshold used in most simulations here). The green cells are those not discarded using a threshold of 30 log-likelihood units. In a), the divergence between the two sequences is $t = 0.05$ expected substitutions per site and the indel rate is $r = 0.1$; the same is in b), but now the central 250 residues in one of the sequences have been removed. In c), the divergence is $t = 0.4$ and the indel rate $r = 0.5$; similarly in d), but again the central 250 residues from one of the two simulated sequences have been removed. e, f) Proportions of columns that are wrongly inferred to be homologous, relative to the number of extant ancestral residues, from simulations similar to the "150 species pairs" simulations when additionally removing parts of sequences (either 100 or 250 residues, either from the beginning, the middle, or the end of a sequence) to simulate missing data. Each pair of neighboring box plots of the same color corresponds to results using an adaptive banding threshold of 15 log-likelihood units (left box in each pair) or 30 log-likelihood units (right box in each pair).

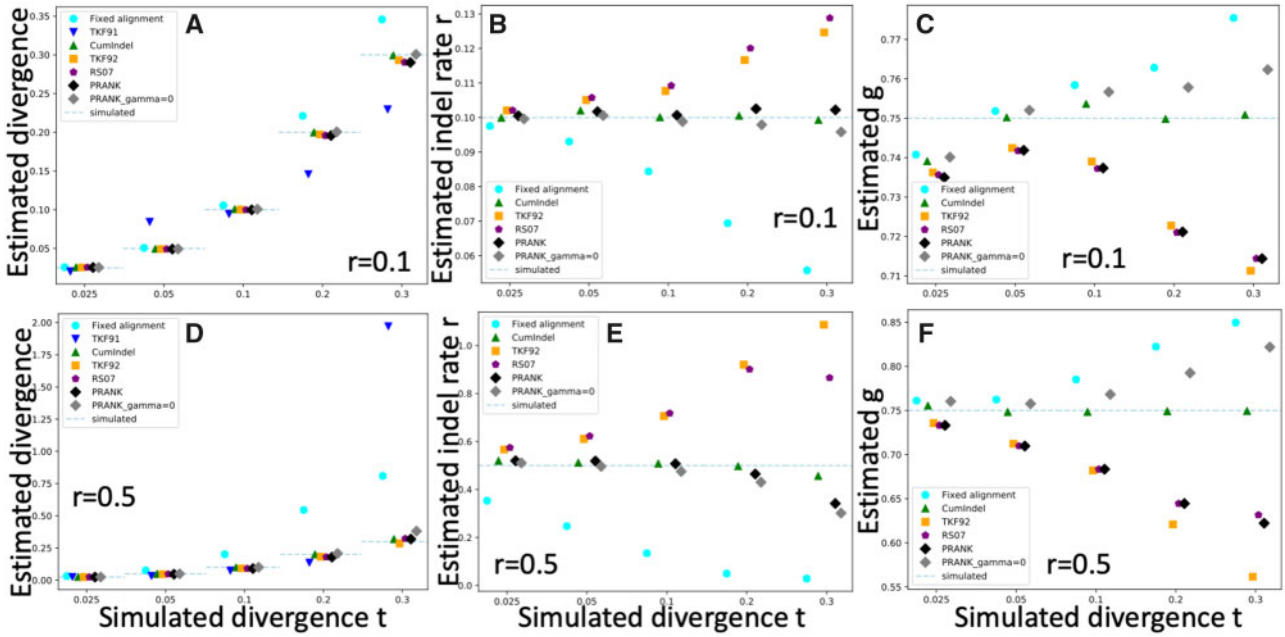

FIGURE 9. Parameter inference in the "1 species pair simulation scenario. Each plot shows inference of one parameter: a,d) divergence time t, b,e) indel rate r, c,f) instantaneous gap extension probability g. For a–c), the simulated indel rate is $r=0.1$, for d–f), it is $r=0.5$. The simulated g is always 0.75, and the ancestral sequence length always 1 kb. The X-axes show the simulated divergence time between each species and its ancestor (so half of the divergence time between the two sequences). For each combination of parameter values, only one inference was performed from a simulated data set of 150 pairwise alignments of the same divergence. Azure horizontal dashed lines show "true" simulated values. "Fixed alignment" refers to values inferred by first estimating EMBOSS alignments, and then inferring parameters on those fixed alignments. All other estimates are from the Forward pairHMM algorithm. In PRANK, the match block extension probability γ is set either to 0 (grey) or equal to the indel block extension probability ($\gamma=\epsilon$, black).

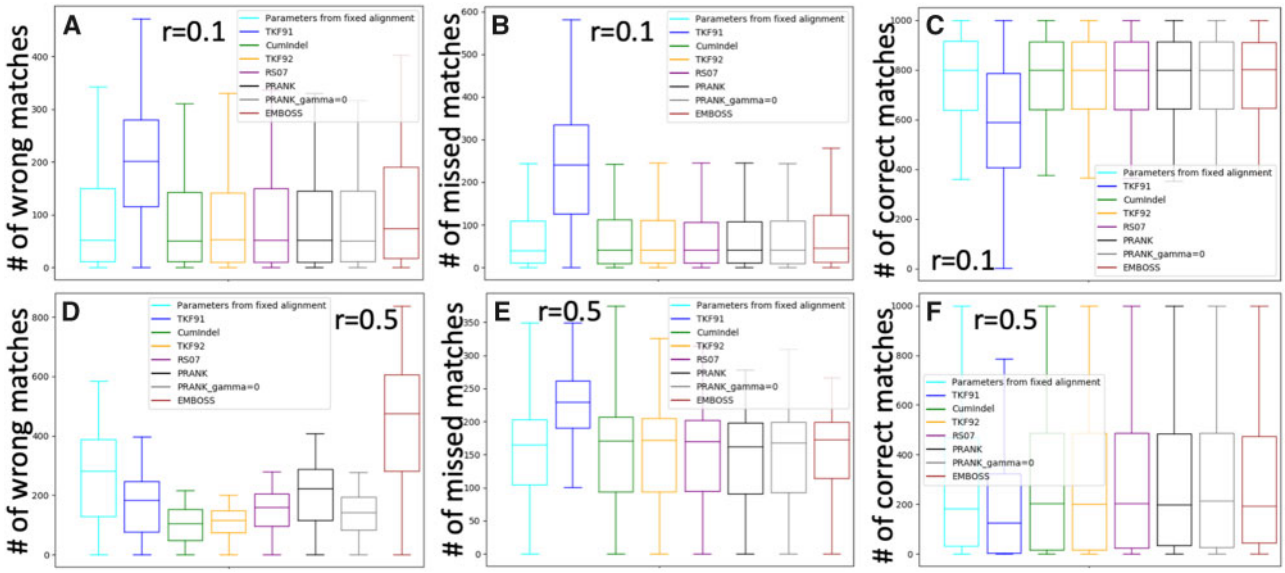

FIGURE 10. **Alignment inference accuracy in the "150 species pairs" simulation scenario.** Performance of different alignment methods when simultaneously modeling different levels of divergence. **A,D** number of wrong match columns per inferred alignment (homology false positives). **B,E** number of non-matched ("missed") simulated homologous residues per alignment (homology false negatives). **C,F** correct match columns per alignment (homology true positives). **A-C** are simulated with $r=0.1$, while **D-F** with $r=0.5$. Each boxplot includes 150 values, corresponding to the 150 pairwise alignments simulated in each scenario. "Cumindel" represents the cumulative indel model. "Parameters from fixed alignment" represents the cumulative indel pairHMM inference with parameter values as inferred from fixed alignments. In PRANK, the match block extension probability γ is set either to 0 (grey) or equal to the indel block extension probability ($\gamma=\epsilon$, black).

UCSC Genome Browser were inferred using LASTZ (Harris 2007) with a gap opening penalty of 600 and a gap extension penalty of 150. From all the synteny blocks, I selected the longest one, which aligns human chromosome 2 (starting at position 143,012,077 with positive orientation) and chimp chromosome 2B (starting at position 29,588,967 with positive orientation). The human sequence in this block is 530,612 bp long, the chimp sequence 530,236 bp, and their alignment inferred by LASTZ is 531,984 columns with 3120 gapped columns and 6196 substitutions.

Joint inference of parameters t (human-chimp divergence), r (indel rate relative to substitutions), g (instantaneous gap extension probability), and κ (transition to transversion rate ratio) was performed under the cumulative indel model and HKY85 substitution model with nucleotide frequencies fixed to observed proportions ($\pi_A = 0.314$, $\pi_C = 0.179$, $\pi_G = 0.182$, and $\pi_T = 0.325$). Inference required between 2 and 10 h on a 2017 MacBook Pro, depending on the starting values, and converged to parameters $t = 0.0118$, $r = 0.0657$, $g = 0.759$, and $\kappa = 4.50$. Due to probably the large numbers of variable tandem repeats between the two considered sequences, adaptive banding required a much stricter threshold than in simulations (Supplementary Fig. S10 available on Dryad). For the results presented here, I used a threshold of 100 log-probability units.

Inferred values are similar to previous studies: divergence is close to the genome-scale 1.24% (Mikkelsen et al. 2005); the inferred indel rate is close to the one inferred in (Cartwright, 2008) from intron alignments. The exception is average instantaneous indel length, here inferred to be 4.15bp, which is much smaller than average indel length inferred in (Cartwright, 2008) ($\approx$ 50), and average indel length observed genome-wide in (Mikkelsen et al., 2005) ($\approx$18). This is probably due to the high synteny of the specific block considered (the average indel length in the UCSC Genome Browser alignment of the same block is 4.14 bp).

I used these inferred parameter values to perform whole-block alignment inference under the cumulative indel model. An adaptive banding threshold of 50 log-probability units or lower resulted in suboptimal alignment due to an average length tandem duplication in chimp that is not present in the human sequence (Supplementary Figs. S11 and S12 available on Dryad). Alignment inference with a threshold between 55 and 80 log-probability units fixed this issue and took between 10 and 20 min to complete.

The cumulative indel alignment differs from the LASTZ alignment in 14 regions (see complete list of local alignments in Supplementary Figs. S13 and S14 available on Dryad). Overall, the cumulative indel alignment has more cumulative indels (763 vs. 753) and more gap columns (3246 vs. 3120), but fewer substitutions (6145 vs. 6196). In one particular region, 10 transversions, 6 transitions, and 1 indel in the LASTZ alignment were replaced by 2 transitions and 2 indels in the cumulative indel alignment (Supplementary Fig. S14, second region). In all but one region, the cumulative indel alignment has more gap columns but fewer substitutions than the LASTZ alignment. The only exception is a region where the cumulative indel model alignment has 1 fewer cumulative indel but 4 more gaps.

An interesting observation is that in four regions, differently from LASTZ, the cumulative indel pairHMM inference places a non-empty cumulative insertion right next to a non-empty cumulative deletion. This reflects an important aspect of the model that is not captured by most non-evolutionary alignment models: non-empty cumulative insertions and deletions are expected next to each other more often than if they were randomly distributed (with parameter values considered here, $\approx$2.85 times more) as also seen in simulations (see variable P^t_{id} in Supplementary Fig. S2 available on Dryad).

Discussion

I presented a new evolutionary indel model, the cumulative indel model. The model can be represented as a finite pairHMM, allowing efficient dynamic programming evolutionary parameter and alignment inference. The cumulative indel model approximates the complex and realistic features of general evolutionary indel models, like the long indel model (Miklós et al. 2004), resulting in high alignment accuracy and interpretability of parameter estimates. Using simulations, I show that most pairHMMs, including the cumulative indel pairHMM, seem to typically outperform traditional score-based aligners. One plausible reason is that constant scores do not accommodate different alignment patterns expected at different levels of divergence. By allowing efficient parameter inference, pairHMMs can improve alignment, and in particular reduce "over-alignment." One exception is the TKF91 pairHMM. This model has been often adopted due to its remarkable mathematical and computational advantages. However, here I show that the TKF91 often leads to worse alignments than classical score-based aligners, and for this reason I do not recommend this model unless indel events are expected to mostly affect only one residue at the time. In fact, TKF91 assumes 1-residue indel events, which is similar to using a linear gap cost, and this assumption probably causes most of its underperformance. By extension, one would expect similar trends from methods that model indels independently across columns (e.g., McGuire et al. 2001; Rivas and Eddy 2008).

A feature that distinguishes the cumulative indel pairHMM from previous finite pairHMMs is that it does not allow transitions from insertion to deletion columns. This constraint simplifies the interpretation of alignments and paths: for example, a transition from a match state to a deletion state represents the presence of a nonempty cumulative deletion, while transitions from a match state to either another match state or an insertion state, both imply the presence of an empty cumulative

deletion. With this approach, a chop zone is represented by only one alignment, and this makes it easier to translate parameters from cumulative indel distributions into a finite pairHMM, and to interpret the meaning of an alignment in evolutionary terms. However, the main advantage of this feature is that it reduces the number of possible alignments to be considered. In pairwise alignments, this can only lead to modest computational savings, as only one pairHMM state transition out of a total of nine is prohibited. However, this could lead to more substantial improvements in computational demand when considering multiple sequence alignment and treeHMMs. One drawback of this assumption is however that it reduces the numbers of parameters of the pairHMM, and in at least some cases it can reduce its potential realism. For example, under the assumptions of the TKF91 model, the TKF91 describes the distribution of chop zones in the GIM exactly. In the same scenario, the cumulative indel model ignores the correlation between nonempty cumulative deletion length and nonempty cumulative insertion length, and therefore it represents only an approximation. In the future, it would be interesting to try to improve the realism of the cumulative indel model by using the transition probability from insertion state to deletion state as an additional parameter to account for this correlation.

Here, I also presented "adaptive banding," a technique to reduce computational demand of pairHMMs and dynamic programming alignment in general. Adaptive banding is particularly effective with recently diverged sequences, where it only explores a small band of the dynamic programming alignment matrix. For highly diverged sequences or regions of low homology, adaptive banding automatically increasing the size of the band, accommodating for the global or local higher alignment uncertainty. This technique does not rely on prior pseudo-optimal alignments, and generally does not reduce accuracy, except in the presence of long duplications, long indels and long missing sequence parts. As shown here with the statistical alignment and parameter inference of a large (≈530kb) human-chimp synteny block, adaptive banding paves the way to genome-wide statistical alignment. In future, it could be possible to further reduce computational demand of the presented methods using C and adapting some of the strategies in HMMoC (Lunter 2007a).

Future Applications

The cumulative indel model and adaptive banding have a number of possible applications beyond pairwise alignment and evolutionary parameter inference. One natural extension of this work is to multiple sequence alignment. For example, the presented techniques could be included in phylogenetically aware multiple sequence aligners such as PRANK (Löytynoja and Goldman 2005, 2008a, 2010) and could also be used in treeHMMs such as BAli-Phy (Redelings and Suchard 2005, 2007). Another application could be in iterative methods of phylogeny and alignment search (Liu et al. 2009, 2011; Mirarab et al. 2015).

(Bogusz and Whelan, 2017) used statistical pairwise alignment to calculate distance matrices, and, in turn, phylogenetic trees. The cumulative indel model and adaptive banding could also be used in this context, for fast phylogenetic, guide tree, or alignment parameter inference.

Future Extensions

Evolutionary indel models typically aim at describing sequence evolution in neutral settings. Selective pressure, and in particular structural constraints, can have dramatic effects on indel distributions, causing clusters of indels and substitutions in some areas of the alignment, while leaving others more stable. Modeling nonuniform distributions of evolutionary events can therefore lead better alignments (Löytynoja and Goldman 2008a). However, it is still an open question how to efficiently and accurately account for selection in alignment inference. Simulating sequence evolution in this context is also an active filed of research (Koestler et al. 2012). Selection could, for example, be included in the cumulative indel model by allowing stretches of invariable sites, or adding hidden states describing regions under purifying selection.

Another open question is to how best model codon alignments. Codon alignments, in fact, carry the complication that indels can affect the codon frame. To simplify this problem, aligners and simulators typically (and unrealistically) assume that indels in codon alignments can only insert or delete entire codons. This assumption can lead to biases, for example overestimating the number of substitutions (Redelings and Suchard 2007). It could be possible to address this issue by extending the state space of the cumulative indel pairHMM (see Hein 1994; Arvestad 1997; Pedersen et al. 1998), but at the cost of additional computational demand.

It could also be possible to rephrase the cumulative indel model as a homology structure (Lunter et al. 2005) or alignment graph (Herman et al. 2015) model. These structures have the advantage of not requiring ordering of exchangeable alignment columns.

In the future, I also plan to address the issues that adaptive banding faces with large duplications. It could be possible to detect this, and possibly other alignment issues, by running adaptive banding in both directions and investigate differences in the two alignments. A similar bidirectional approach would also be key for a version of adaptive banding with further reduced memory demand, similar to the Hirschberg algorithm (Hirschberg 1975). It is however less clear how to address the problems faced by adaptive banding when large portions of sequences are missing. While, in these scenarios, increasing the log-likelihood threshold helps,

it is usually not possible to choose a threshold that would allow missing sequence parts of any arbitrary length. In alignments of moderate size, such as protein alignments, a threshold could be chosen that is known to be robust to missing sequence parts of the order of magnitude of the considered sequence lengths. For example, a threshold of 30 log-likelihood units seems to robust, in our simulations, to missing sequence parts of 250 residues, at realistic levels of divergence, without sacrificing most of the computational benefits of adaptive banding. However, another approach that could be promising would be, when possible, to combine adaptive banding with other techniques to guide the alignment, for example using k-mer matches for anchoring the alignment (see e.g., Bogusz and Whelan 2017).

Conclusion

The cumulative indel model represents a promising approach to improve pairwise alignment efficiency and accuracy. Adaptive banding allows further reductions in statistical alignment costs. For these reasons, the methods presented here have the potential to importantly impact future alignment, phylogenetic, and molecular evolution inference.

Supplementary Material

Supplementary Figures and Text are available from the Dryad Digital Repository: http://dx.doi.org/10.5061/dryad.rbnzs7h8m.

Acknowledgments

I am extremely thankful to Nick Goldman for supporting me and for his help with the preparation of the manuscript. I also thank Benjamin Redelings, Conor Walker, Ian Holmes, and Jotun Hein for helpful comments and corrections on early drafts of the manuscript.

References

Arvestad L. 1997. Aligning coding DNA in the presence of frame-shift errors. In: Apostolico A., Hein J., editors. Combinatorial Pattern Matching. CPM 1997. Lecture Notes in Computer Science, Vol. 1264. Berlin, Heidelberg: Springer.

Bogusz M., Whelan S. 2017. Phylogenetic tree estimation with and without alignment: new distance methods and benchmarking. Syst. Biol. 66(2): 218–231.

Bouchard-Côté A., Klein D., Jordan M.I. 2009. Efficient inference in phylogenetic indel trees. In: Koller D., Schuurmans D., Bengio Y., Bottou L., editors. Advances in neural information processing systems. Red Hook (NY): Curran Associates, Inc. p. 177–184.

Bressert E. 2012. SciPy and NumPy: an overview for developers. Sebastopol (CA): O'Reilly Media, Inc.

Cartwright R.A. 2005. DNA assembly with gaps (DAWG): simulating sequence evolution. Bioinformatics 21(Suppl_3):iii31–iii38.

Cartwright R.A. 2008. Problems and solutions for estimating indel rates and length distributions. Mol. Biol. Evol. 26(2):473–480.

Chao K.-M., Pearson W.R., Miller W. 1992. Aligning two sequences within a specified diagonal band. Bioinformatics 8(5):481–487.

Durbin R., Eddy S.R., Krogh A., Mitchison G. 1998. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge: Cambridge University Press.

Fleissner R., Metzler D., Von Haeseler A. 2005. Simultaneous statistical multiple alignment and phylogeny reconstruction. Syst. Biol. 54(4):548–561.

Fletcher W., Yang Z. 2009. Indelible: a flexible simulator of biological sequence evolution. Mol. Biol. Evol. 26(8):1879–1888.

Gao F., Han L. 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. Comput. Optim. Appl. 51(1): 259–277.

Gregory T.R. 2004. Insertion–deletion biases and the evolution of genome size. Gene 324:15–34.

Harris R.S. 2007. Improved pairwise alignment of genomic DNA [PhD thesis].

Hasegawa M., Kishino H., Yano T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol. 22(2):160–174.

Havgaard J.H., Torarinsson E., Gorodkin J. 2007. Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. PLoS Comput. Biol. 3(10):e193.

Hein J. 1994. An algorithm combining DNA and protein alignment. J. Theor. Biol. 167(2):169–174.

Hein J. 2000. An algorithm for statistical alignment of sequences related by a binary tree. In: Altman R.B., Dunker A.K., Hunker L., Lauderdale K., Klein T.E., editors. Biocomputing 2001. Singapore: World Scientific. p. 179–190.

Hein J., Wiuf C., Knudsen B., Møller M., Wibling G. 2000. Statistical alignment: computational properties, homology testing and goodness-of-fit. J. Mol. Biol. 302(1):265–279.

Herman J.L., Novák Á., Lyngsø R., Szabó A., Miklós I., Hein J. 2015. Efficient representation of uncertainty in multiple sequence alignments using directed acyclic graphs. BMC Bioinformatics 16(1):108.

Hirschberg D.S. 1975. A linear space algorithm for computing maximal common subsequences. Commun. ACM 18(6):341–343.

Holmes I., Bruno W.J. 2001. Evolutionary HMMs: a Bayesian approach to multiple alignment. Bioinformatics, 17(9):803–820.

Holmes I.H. 2017. Solving the master equation for indels. BMC Bioinformatics 18(1):255.

Iantorno S., Gori K., Goldman N., Gil M., Dessimoz C. 2014. Who watches the watchmen? An appraisal of benchmarks for multiple sequence alignment. In: Russell D, editor. Multiple sequence alignment methods. Methods in Molecular Biology (Methods and Protocols), Vol. 1079. Totowa (NJ): Humana Press.

Koestler T., von Haeseler A., Ebersberger I. 2012. Revolver: modeling sequence evolution under domain constraints. Mol. Biol. Evol. 29(9):133–2145.

Le S.Q., Gascuel O. 2008. An improved general amino acid replacement matrix. Mol. Biol. Evol. 25(7):1307–1320.

Levy Karin E., Ashkenazy H., Hein J., Pupko T. 2019. A simulation-based approach to statistical alignment. Syst. Biol. 68(2):252–266.

Liu K., Raghavan S., Nelesen S., Linder C.R., Warnow T. 2009. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science 324(5934):1561–1564.

Liu K., Warnow T.J., Holder M.T., Nelesen S.M., Yu J., Stamatakis, A.P., Linder C.R. 2011. SATe-II: very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. Syst. Biol. 61(1):90–106.

Löytynoja A., Goldman N. 2005. An algorithm for progressive multiple alignment of sequences with insertions. Proc. Natl. Acad. Sci. USA 102(30):10557–10562.

Löytynoja A., Goldman N. 2008a. A model of evolution and structure for multiple sequence alignment. Philos. Trans. R. Soc. Lond. B 363(1512):3913–3919.

Löytynoja A., Goldman N. 2008b. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. Science 320(5883):1632–1635.

Löytynoja A., Goldman N. 2010. webPRANK: a phylogeny-aware multiple sequence aligner with interactive alignment browser. BMC Bioinformatics 11(1):579.

Lunter G. 2007a. Hmmoc—a compiler for hidden Markov models. Bioinformatics 23(18):2485–2487.

Lunter G. 2007b. Probabilistic whole-genome alignments reveal high indel rates in the human and mouse genomes. Bioinformatics 23(13):i289–i296.
Lunter G., Miklós I., Drummond A., Jensen J.L., Hein J. 2005. Bayesian coestimation of phylogeny and sequence alignment. BMC Bioinformatics 6(1):83.
McGuire G., Denham M.C., Balding D.J. 2001. Models of sequence evolution for DNA sequences containing gaps. Mol. Biol. Evol. 18(4):481–490.
Metzler D. 2003. Statistical alignment based on fragment insertion and deletion models. Bioinformatics 19(4):490–499.
Metzler D., Fleißner R., Wakolbinger A., von Haeseler A. 2001. Assessing variability by joint sampling of alignments and mutation rates. J. Mol. Evol. 53(6):660–669.
Mikkelsen T., Hillier L., Eichler E., Zody M., Jaffe D., Yang S.-P., Enard W., Hellmann I., Lindblad-Toh K., Altheide T., et al. 2005. Initial sequence of the chimpanzee genome and comparison with the human genome. Nature 437(7055):69–87.
Miklós I., Lunter G., Holmes I. 2004. A long indel model for evolutionary sequence alignment. Mol. Biol. Evol. 21(3):529–540.
Miklós I., Novák Á., Satija R., Lyngsø R., Hein J. 2009. Stochastic models of sequence evolution including insertion–deletion events. Stat. Methods Med. Res. 18(5):453–485.
Mirarab S., Nguyen N., Guo S., Wang L.-S., Kim J., Warnow T. 2015. Pasta: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. J. Comput. Biol. 22(5):377–386.
Mitchison G. 1999. A probabilistic treatment of phylogeny and sequence alignment. J. Mol. Evol. 49(1):11–22.
Mitchison G., Durbin R. 1995. Tree-based maximal likelihood substitution matrices and hidden Markov models. J. Mol. Evol. 41(6):1139–1151.
Newman T.L., Tuzun E., Morrison V.A., Hayden K.E., Ventura M., McGrath S.D., Rocchi M., Eichler E.E. 2005. A genome-wide survey of structural variation between human and chimpanzee. Genome Res. 15(10):1344–1356.
Notredame C. 2007. Recent evolutions of multiple sequence alignment algorithms. PLoS Comput. Biol. 3(8):e123.
Novák Á., Miklós I., Lyngsø R., Hein J. 2008. Statalign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees. Bioinformatics 24(20):2403–2404.
Pedersen C.N., Lyngsø R., Hein J. 1998. Comparison of coding DNA. In: Farach-Colton M., editor. Combinatorial Pattern Matching. CPM 1998. Lecture Notes in Computer Science. Vol. 1448. Berlin, Heidelberg: Springer.
Redelings B.D., Suchard M.A. 2005. Joint Bayesian estimation of alignment and phylogeny. Syst. Biol. 54(3):401–418.
Redelings B.D., Suchard M.A. 2007. Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. BMC Evol. Biol. 7(1):40.
Rice P., Longden I., Bleasby A. 2000. Emboss: the european molecular biology open software suite. Trends Genetics 16(6):276–277.
Rivas E., Eddy S.R. 2008. Probabilistic phylogenetic inference with insertions and deletions. PLoS Comput. Biol. 4(9):e1000172.
Rivas E., Eddy S.R. 2015. Parameterizing sequence alignment with an explicit evolutionary model. BMC Bioinformatics 16(1): 406.
Rosenberg M.S. 2005. Myssp: non-stationary evolutionary sequence simulation, including indels. Evol. Bioinformatics 1:117693430500100007.
Strope C.L., Abel K., Scott S.D., Moriyama E.N. 2009. Biological sequence simulation for testing complex evolutionary hypotheses: indel-seq-gen version 2.0. Mol. Biol. Evol. 26(11):2581–2593.
Suzuki H., Kasahara M. 2017. Acceleration of nucleotide semi-global alignment with adaptive banded dynamic programming. BioRxiv. New York: Cold Spring Harbor Laboratory. p. 130633.
Tan G., Gil M., Löytynoja A.P., Goldman N., Dessimoz C. 2015. Simple chained guide trees give poorer multiple sequence alignments than inferred trees in simulation and phylogenetic benchmarks. Proc. Natl. Acad. Sci. USA 112(2):E99–E100.
Thorne J.L., Kishino H., Felsenstein J. 1991. An evolutionary model for maximum likelihood alignment of DNA sequences. J. Mol. Evol. 33(2):114–124.
Thorne J.L., Kishino H., Felsenstein J. 1992. Inching toward reality: an improved likelihood model of sequence evolution. J. Mol. Evol. 34(1):3–16.
Westesson O., Barquist L., Holmes I. 2012. Handalign: Bayesian multiple sequence alignment, phylogeny and ancestral reconstruction. Bioinformatics 28(8):1170.
Yang Z., Rannala B. 2012. Molecular phylogenetics: principles and practice. Nat. Rev. Genetics 13(5):303.