Progressive Multiple Sequence Alignment with the Poisson Indel Process

Massimo Maiolo[1], Xiaolei Zhang[2], Manuel Gil[1], and Maria Anisimova[*1]

[1]*Institute of Applied Simulation, School of Life Sciences and Facility Management, Zurich University of Applied Sciences (ZHAW), Wädenswil, Switzerland*

[2]*National Heart and Lung Institute, Imperial College London, London, United Kingdom*

Abstract

Sequence alignment lies at the heart of many evolutionary and comparative genomics studies. However, the optimal alignment of multiple sequences is NP-hard, so that exact algorithms become impractical for more than a few sequences. Thus, state of the art alignment methods employ progressive heuristics, breaking the problem into a series of pairwise alignments guided by a phylogenetic tree. Changes between homologous characters are typically modelled by a continuous-time Markov substitution model. In contrast, the dynamics of insertions and deletions (indels) are not modelled explicitly, because the computation of the marginal likelihood under such models has exponential time complexity in the number of taxa. Recently, Bouchard-Côté and Jordan [PNAS (2012) 110(4):1160–1166] have introduced a modification to a classical indel model, describing indel evolution on a phylogenetic tree as a Poisson process. The model termed PIP allows to compute the joint marginal probability of a multiple sequence alignment and a tree in linear time. Here, we present an new dynamic programming algorithm to align two multiple sequence alignments by maximum likelihood in polynomial time under PIP, and apply it a in progressive algorithm. To our knowledge, this is the first progressive alignment method using a rigorous mathematical formulation of an evolutionary indel process and with polynomial time complexity.

1 Introduction

Multiple sequence alignments (MSAs) are routinely required in the early stages of comparative and evolutionary genomics studies. Not surprisingly,

*Corresponding author: maria.anisimova@zhaw.ch

accuracy of MSA inference affects subsequent analyses that rely on an MSA estimates [13]. MSA estimation is among the oldest bioinformatics problems, yet remains intensely studied due to its complexity (NP-hard [5, 1, 12]). The progressive alignment approach has allowed to reduce the overall computational complexity to polynomial time by breaking the MSA problem into a series of pairwise alignments guided by a tree representing the evolutionary relationship of sequences. Today most popular alignment programs employ the progressive approach (e.g., ClustalW [10], MAFFT [6], MUSCLE [4], PRANK [7] and T-Coffee [3, 9] among others).

All state of the art MSA programs nowadays use an evolutionary model to describe changes between homologous characters, providing a more realistic description of molecular data and thus more accurate inferences. However, a mathematical formulation of the insertion-deletion (indel) process still remains a critical issue. Describing the indel process in probabilistic terms is more challenging: unlike substitutions, indels often involve several sites, vary in length and may overlap obscuring the underlying mechanisms. Instead, the popular PRANK program adopts a pragmatic approach; it uses an outgroup to distinguish insertions from deletions during the progressive alignment procedure, so that they are penalised differently [8]. As a result, PRANK produces exceptionally accurate alignments, notably with densely sampled data and given an accurate guide tree. Still the method lacks a mathematical model describing the evolution of indels. Indeed, the computation of the marginal likelihood under the classical indel model TKF91 [11] is exponential in the number of taxa due to the absence of site independence assumption.

A recent modification of the TKF91 describes the evolution of indels on a phylogenetic tree as a Poisson process, thus dubbed the Poisson indel process or the PIP model [2]. Consequently, standard mathematical results, particularly the Poisson thinning, allow to achieve linear time complexity for computing the joint marginal probability of a tree and an MSA. This includes analytic marginalisation of unobservable homologous paths which occur whenever an ancestral character is inserted and subsequently deleted, and consequently cannot be detected in the extant sequences. For a given MSA and a tree, a likelihood score under PIP can be computed in linear time. This score can be used to find the maximum a posteriori tree-alignment solution. Remarkably, this breakthrough allows for a necessary rigorous way of combining models of substitutions and indels, and a tractable computation of the marginal likelihood function. Nevertheless, at the moment the algorithm has been applied in a Bayesian framework via tree-alignment space sampling.

Here we propose a new progressive algorithm to estimate an MSA under the explicit model with substitutions and indels. We have re-framed the original PIP equations into a dynamic programming (DP) approach. It aligns two MSAs (represented by their homology paths on the tree) by maximum

likelihood (ML) in polynomial time. The progressive algorithm traverses a guide tree in post order; at each internal node the DP is applied to align the two sub-alignments at the child nodes. The procedure terminates at the root of the guide tree, with the complete MSA and the corresponding likelihood, which by construction is the likelihood under the PIP model. We have implemented the progressive MSA algorithm in a prototype program and verified its correctness by simulation. To our knowledge, this is the first progressive MSA algorithm using a rigorous mathematical formulation of an indel process and with polynomial time complexity.

The remainder of this manuscript is organized as follows. We first introduce notation and the PIP model. Then, we describe our DP algorithm and provide the simulation results.

2 The PIP model

Let $\tau = (\mathcal{V}, \mathcal{E}, b)$ represent a rooted binary phylogenetic tree with N leaves. τ is a directed, connected, labelled acyclic graph, with a finite set of branching points $\mathcal{V}$ of cardinality $|\mathcal{V}| = 2N - 1$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Leaves $\mathcal{L} \subset \mathcal{V}$ denotes N observed taxa, represented by strings of characters from a finite alphabet Σ (nucleotides, amino acids or codons). There are $N - 1$ internal vertices $v \subset \mathcal{V}$ whereof the root Ω is the most recent common ancestor of all leaves. Branch length $b(v)$ associated with node $v \in \mathcal{V}$ spans from v to its parent node $\mathtt{pa}(v)$. The total tree length $\|\tau\|$ is a sum of all the branch lengths.

The PIP model describes a string-valued evolutionary process along the branches of τ. We denote the Lebesgue measure on τ, i.e. the distance from the root to a given point on the tree, by the same symbol τ. Atomic insertions are Poisson events with rate measure $\nu(\mathrm{d}t) = \lambda(\tau(\mathrm{d}t) + \mu^{-1}\delta_\Omega(\mathrm{d}t))$, where λ is the insertion rate, μ the deletion rate, and $\delta_\Omega(\cdot)$ Dirac's delta function, which is one at the root Ω and zero everywhere else. This formulation guarantees that the expected sequence length remains constant during the whole evolutionary process. Point substitutions and deletions are modelled by a continuous-time Markov process on $\Sigma_\epsilon = \Sigma \cup \epsilon$, where ϵ is the deletion (or gap) symbol. Accordingly, the process generator matrix $\mathbf{Q}_\epsilon$ of the combined substitution and indel process extends the instantaneous substitution rate matrix $\mathbf{Q}$ by a row and a column to include ϵ, which is modelled as an absorbing state as there can be no substitutions after a deletion event. The quasi-stationary distribution of $\mathbf{Q}_\epsilon$ is denoted by π^ϵ . Root Ω has a virtual infinite length stem, reflecting the equilibrium steady state distribution of the characters at the root.

The probability $\iota(v)$ of inserting a single character on branch $\mathrm{pa}(v) \to v$ is proportional to branch length $b(v)$. For $v \neq \Omega$ it is given by $\iota(v) = b(v)/(\|\tau\|+\mu^{-1})$; at the root atomic mass point probability $\iota(\Omega) = \mu^{-1}/(\|\tau\|+$

μ^{-1}) so that $\sum_{v\in\mathcal{V}} \iota(v) = 1$. The survival probability $\beta(v)$ associated with an inserted character on branch $\mathrm{pa}(v) \to v$ is given by $\beta(\Omega) = 1$ and $\beta(v) = (1 - \exp(-\mu b(v)))/(\mu b(v))$.

The marginal likelihood $p_\tau(m)$ of MSA m of length $|m|$ is computable in $O(N \cdot |m|)$ and can be expressed as

$$p_\tau(m) = \varphi(p(c_\emptyset), |m|) \prod_{c\in m} p(c), \tag{1}$$

where $p(c)$ is the likelihood of a single column c, and $p(c_\emptyset)$ is the likelihood of an unobservable character history, represented by a column $c_\emptyset$ with a gap at every leaf. The factor in (1)

$$\varphi(p(c_\emptyset), |m|) = \|\nu\|^{|m|} \exp(-\|\nu\|(p(c_\emptyset) - 1))/|m|! \tag{2}$$

is the marginal likelihood over all unobservable character histories, where $\|\nu\|$ is the normalising Poisson intensity.

The column likelihood can be expressed as $p(c) = \sum_{v\in\mathcal{V}} \iota(v) f_v$, where f_v denotes the probability of c, given that the corresponding character was inserted at v. This probability can be computed in $O(N)$ using a variant of Felsenstein's peeling recursion. Let S be the set of leaves that do not have a gap in column c, and A the set of nodes ancestral to S. Then

$$f_v = \begin{cases} \tilde{f}_v & \text{if } v = \Omega \\ \mathbf{1}[v \in \mathrm{A}]\beta(v)\tilde{f}_v & \text{if } c \neq c_\emptyset \text{ and } v \neq \Omega \\ 1 + \beta(v)\left(-1 + \tilde{f}_v\right) & \text{otherwise (i.e. if } c = c_\emptyset \text{ and } v \neq \Omega), \end{cases} \tag{3}$$

where

$$\tilde{f}_v = \sum_{\sigma\in\Sigma} \pi^\epsilon(\sigma) \tilde{f}_v(\sigma)$$

and

$$\tilde{f}_v(\sigma) = \begin{cases} \mathbf{1}[c(v) = \sigma] & \text{if } v \in \mathcal{L} \\ \sum_{\sigma'\in\Sigma_\epsilon} \exp(b(v)\mathbf{Q}_\epsilon)_{\sigma,\sigma'} \prod_{w\in\mathrm{child}(v)} \tilde{f}_w(\sigma') & \text{otherwise,} \end{cases}$$

and $\mathbf{1}[\cdot]$ is the indicator function.

3 Dynamic programming under PIP

Given an internal node v, our DP algorithm proceeds to align the two sub-alignments obtained in the left and right sub-trees maximizing the likelihood (Eq. 1) of the tree rooted at v. Let $\mathbf{X}$ and $\mathbf{Y}$ denote these sub-alignments, respectively with $N_\mathbf{X}$ and $N_\mathbf{Y}$ sequences and alignment lengths $|\mathbf{X}|$ and $|\mathbf{Y}|$.

If a sub-tree is a leaf then the sub-alignment, say $\mathbf{X}$, is reduced to an input sequence, i.e. $N_{\mathbf{X}} = 1$ and $|\mathbf{X}|$ corresponds the sequence length.

Note that the marginal likelihood function $p_\tau(m)$ (Eq. 1) is not monotonically increasing in the alignment length $|m|$. While the product of column likelihoods is monotonically increasing, the marginal likelihood of unobserved histories $\varphi(p(c_\emptyset), |m|)$ is non-monotonic. This means that $p_\tau(m)$ cannot be maximised by means of a standard two-dimensional DP approach (in particular, because the alignment length is not known a priori). Our algorithm accounts for the dependence on alignment length with a third dimension.

The algorithm works with three three-dimensional sparse matrices $\mathbf{S}^{\mathrm{M}}$, $\mathbf{S}^{\mathrm{X}}$ and $\mathbf{S}^{\mathrm{Y}}$ each of size $(|\mathbf{X}|+1)\times(|\mathbf{Y}|+1)\times(|\mathbf{X}|+|\mathbf{Y}|)$ with entries defined as follows:

1. *match* cell $\mathbf{S}^{\mathrm{M}}_{i,j,k}$ contains the likelihood of the partial optimal MSA between $\mathbf{X}_1 \ldots \mathbf{X}_i$ and $\mathbf{Y}_1 \ldots \mathbf{Y}_j$ of length k with the columns $\mathbf{X}_i$ and $\mathbf{Y}_j$ aligned. Consequently, all characters in the two columns are inferred to be homologous.

2. *gapX* cell $\mathbf{S}^{\mathrm{X}}_{i,j,k}$ contains the likelihood of the partial optimal MSA between $\mathbf{X}_1 \ldots \mathbf{X}_i$ and $\mathbf{Y}_1 \ldots \mathbf{Y}_j$ of length k with the column $\mathbf{X}_i$ aligned with a column of size $N_{\mathbf{Y}}$ containing gaps only. The characters in the two columns do not share a common history, either because the ancestor character had been deleted on the right subtree, or because it had been inserted on the left subtree, below the node v.

3. similarly, *gapY* cell $\mathbf{S}^{\mathrm{Y}}_{i,j,k}$ matches column $\mathbf{Y}_j$ with a column of size $N_{\mathbf{X}}$ containing gaps only.

Forward phase

Each matrix $\mathbf{S}^{\mathrm{M}}$, $\mathbf{S}^{\mathrm{X}}$ and $\mathbf{S}^{\mathrm{Y}}$ is initialized with $\varphi(p(c_\emptyset)), 0)$ at position $(0,0,0)$ and a zero in every other position. The DP equations are:

$$\mathbf{S}^{\mathrm{M}}_{i,j,k} = \frac{\|\nu\|}{k} \cdot p\left(\begin{bmatrix}\mathbf{X}_i \\ \mathbf{Y}_j\end{bmatrix}\right) \cdot \max\left\{\mathbf{S}^{\mathrm{M}}_{i-1,j-1,k-1}, \mathbf{S}^{\mathrm{X}}_{i-1,j-1,k-1}, \mathbf{S}^{\mathrm{Y}}_{i-1,j-1,k-1}\right\}$$
$$\text{for } i = 1,\ldots,|\mathbf{X}| \text{ and } j = 1,\ldots,|\mathbf{Y}| \text{ and } k = 1,\ldots,|\mathbf{X}|+|\mathbf{Y}|, \quad (4)$$

$$\mathbf{S}^{\mathrm{X}}_{i,j,k} = \frac{\|\nu\|}{k} \cdot p\left(\begin{bmatrix}\mathbf{X}_i \\ c_\emptyset\end{bmatrix}\right) \cdot \max\left\{\mathbf{S}^{\mathrm{M}}_{i-1,j,k-1}, \mathbf{S}^{\mathrm{X}}_{i-1,j,k-1}, \mathbf{S}^{\mathrm{Y}}_{i-1,j,k-1}\right\}$$
$$\text{for } i = 1,\ldots,|\mathbf{X}| \text{ and } j = 0,\ldots,|\mathbf{Y}| \text{ and } k = 1,\ldots,|\mathbf{X}|+|\mathbf{Y}|, \quad (5)$$

$$\mathbf{S}^{\mathrm{Y}}_{i,j,k} = \frac{\|\nu\|}{k} \cdot p\left(\begin{bmatrix} c_\emptyset \\ \mathbf{Y}_j \end{bmatrix}\right) \cdot \max\{\mathbf{S}^{\mathrm{M}}_{i,j-1,k-1}, \mathbf{S}^{\mathrm{X}}_{i,j-1,k-1}, \mathbf{S}^{\mathrm{Y}}_{i,j-1,k-1}\}$$
$$\text{for } i = 0, \ldots, |\mathbf{X}| \text{ and } j = 1, \ldots, |\mathbf{Y}| \text{ and } k = 1, \ldots, |\mathbf{X}| + |\mathbf{Y}|. \quad (6)$$

The symbol $c_\emptyset$ in Eq.s 5 and 6 represents a column with gaps, respectively of length $N_{\mathbf{Y}}$ and $N_{\mathbf{X}}$. The factor $\|\nu\|/k$ successively constructs $\varphi(p(c_\emptyset), k)$ along the third dimension as columns are added into partial alignments. Note that the column likelihoods $p(\cdot)$ can be computed in constant time from the corresponding column likelihoods at the two children of v, by re-using appropriate summands (defined by the set A in Eq. 3).

Backtracking

An optimal alignment is determined by backtracking along a matrix $\mathbf{TR}$ of size $(|\mathbf{X}| + 1) \times (|\mathbf{Y}| + 1) \times (|\mathbf{X}| + |\mathbf{Y}|)$. In the forward phase, $\mathbf{TR}$ records at position (i, j, k) the name of the DP matrix ("$\mathbf{S}^{\mathrm{M}}$", "$\mathbf{S}^{\mathrm{X}}$", or "$\mathbf{S}^{\mathrm{Y}}$") with highest likelihood at the same position (i, j, k). If the maximum is not unique then a uniform random choice is made. The backtracking algorithm starts at $\mathbf{TR}(|\mathbf{X}|, |\mathbf{Y}|, k_0)$, where

$$k_0 = \underset{k=\max(|\mathbf{X}|,|\mathbf{Y}|)..(|\mathbf{X}|+|\mathbf{Y}|)}{\arg\max} \left[\mathbf{S}^{\mathrm{M}}(|\mathbf{X}|, |\mathbf{Y}|, k), \mathbf{S}^{\mathrm{X}}(|\mathbf{X}|, |\mathbf{Y}|, k), \mathbf{S}^{\mathrm{Y}}(|\mathbf{X}|, |\mathbf{Y}|, k)\right]$$

is the length of the best scoring alignment. If k_0 is not unique a random uniform choice is made. $\mathbf{TR}$ is then traversed from $(|\mathbf{X}|, |\mathbf{Y}|, k_0)$ to $(0, 0, 0)$. Suppose the algorithm is at position (i, j, k). If $\mathbf{TR}(i, j, k) =$ "$\mathbf{S}^{\mathrm{M}}$" then the columns $\mathbf{X}_i$ and $\mathbf{Y}_j$ are matched and all the indices are decremented, i.e. $i = i - 1$, $j = j - 1$ and $k = k - 1$. If $\mathbf{TR}(i, j, k) =$ "$\mathbf{S}^{\mathrm{X}}$" then the column $\mathbf{X}_i$ is matched with a column of gaps of size $N_{\mathbf{Y}}$ and the indices i and k are decremented, and, if $\mathbf{TR}(i, j, k) =$ "$\mathbf{S}^{\mathrm{Y}}$" then the column $\mathbf{Y}_j$ is matched with a column of gaps of size $N_{\mathbf{X}}$ and the indices j and k are decremented.

4 Empirical verification of correctness

We have implemented our progressive algorithm in a prototype program. To test the correctness of algorithm and implementation, we generated data under PIP using a simulator provided by the authors [1] of PIP. We chose relatively small trees and short sequences to be able to perform analytical tests during algorithm design an program debugging. Specifically, we simulated 120 datasets in total, on trees with 4, 5, 6 and 7 leaves, using all the combinations of $\lambda \in \{0.1, 1\}$ and $\mu \in \{0.1, 1\}$. The resulting sequence-lengths varied between 5 and 8 nucleotides.

[1] personal communication

The simulated data was analyzed with our program using correct model parameters and guide-trees. First, we confirmed the correctness of the likelihoods obtained with the DP algorithm, by scoring the resulting MSAs with an independent implementation provided by the authors of PIP. In all cases the likelihood matched. In a second test, we verified that the DP generates optimal pairwise MSA alignments. To this end, all the possible pairwise alignments were generated at each internal node of the guide-trees and scored with the independent implementation. The DP algorithm always reconstructed an optimal MSA.

5 Conclusion

We have developed and implemented a progressive alignment algorithm that relies on PIP, and, thus, uses a continuous-time Markov model to describe insertions, deletions, and substitutions. The core of our method is a new DP algorithm for the alignment of two MSAs by ML, which exploits PIP's linear time complexity (in the number of taxa and the sequence length) for the computation of marginal likelihoods. The overall complexity of the progressive algorithm is $O(Nl^3)$, where N is number of taxa and l the sequence length. The cubic factor stems from the fact that the likelihood is not monotonically increasing in the MSA length, so that the length has to be incorporated as an extra dimension in the DP. However, empirical findings show that the likelihood has exactly one maximum, suggesting an early stop condition to the DP. We are currently optimising our implementation in this and other time-critical aspects.

Acknowledgement

We thank Alexandre Bouchard-Côté for providing his code to simulate sequences and to compute marginal likelihoods under PIP. This work was supported by the Swiss National Science Foundation (SNF) grant 31003A_157064 to M. Anisimova.

References

[1] Paola Bonizzoni and Gianluca Della Vedova. The complexity of multiple sequence alignment with sp-score that is a metric. *Theoretical Computer Science*, 259(1):63 – 79, 2001.

[2] Alexandre Bouchard-Côté and Michael I. Jordan. Evolutionary inference via the Poisson Indel Process. *Proceedings of the National Academy of Sciences*, 110(4):1160–1166, January 2013.

[3] Paolo Di Tommaso, Sebastien Moretti, Ioannis Xenarios, Miquel Orobitg, Alberto Montanyola, Jia-Ming Chang, Jean-Francois Taly, and Cedric Notredame. T-coffee: a web server for the multiple sequence alignment of protein and rna sequences using structural information and homology extension. *Nucleic Acids Research*, 39(2):W13, 2011.

[4] Robert C. Edgar. Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5(1):113, 2004.

[5] W. Just. Computational complexity of multiple sequence alignment with SP-score. *J. Comput. Biol.*, 8(6):615–623, 2001.

[6] Kazutaka Katoh and Daron M. Standley. Mafft multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772, 2013.

[7] Ari Löytynoja and Nick Goldman. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, 320(5883):1632–1635, 2008.

[8] Ari Löytynoja and Nick Goldman. An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(30):10557–10562, 2005.

[9] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment1. *Journal of Molecular Biology*, 302(1):205 – 217, 2000.

[10] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22(22):4673–4680, Nov 1994.

[11] Jeffrey L. Thorne, Hirohisa Kishino, and Joseph Felsenstein. An evolutionary model for maximum likelihood alignment of dna sequences. *Journal of Molecular Evolution*, 33(2):114–124, 1991.

[12] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

[13] Karen M. Wong, Marc A. Suchard, and John P. Huelsenbeck. Alignment uncertainty and genomic analysis. *Science*, 319(5862):473–476, 2008.