Alignment uncertainty, regressive alignment and large scale deployment

Evan Floden

Tesi Doctoral UPF / 2018

Universitat Pompeu Fabra - Barcelona

DIRECTOR DE LA TESI

Dr. Cédric Notredame

BIOINFORMATICS AND GENOMICS PROGRAMME

CENTRE FOR GENOMIC REGULATION (CRG)

Dedication

You know how it is. You pick up a thesis, flip to the dedication, and find that, once again, the author has dedicated it to someone else and not to you. Not this time. We may not have ever met, had only a glancing acquaintance, are just crazy about each other, haven't seen each other in far too long or are in some way related. Despite that, I trust we will always think fondly of each other.

This one's for you.

Acknowledgements

These past four years have been phenomenal and made possible by some special people. Foremost thanks must go to Cedric for fostering my scientific development, acting as a professional mentor and becoming a trusted friend whom I am deeply indebted to. The lab environment cultivated by Cedric has given me the opportunity to work with many people much more talented than myself. The mark of each and every one of them can be seen in the work within these pages. The CRG provides an eclectic mix of unparalleled characters from which I have drawn endearing friendships and with whom together we have spent many times we'll never remember with people we will never forget. To my parents who have supported me in my educational pursuits in some form or another over the past three decades. And finally Vero, who has in these trying months been the most outstanding mother that Alistair could ever ask for.

Abstract

A multiple sequence alignment (MSA) provides a description of the relationship between biological sequences where columns represent a shared ancestry through an implied set of evolutionary events. The majority of research in the field has focused on improving the accuracy of alignments within the progressive alignment framework and has allowed for powerful inferences including phylogenetic reconstruction, homology modelling and disease prediction. Notwithstanding this, when applied to modern genomics datasets - often comprising tens of thousands of sequences - new challenges arise in the construction of accurate MSA. These issues can be generalised to form three basic problems. Foremost, as the number of sequences increases, progressive alignment methodologies exhibit a dramatic decrease in alignment accuracy. Additionally, for any given dataset many possible MSA solutions exist, a problem which is exacerbated with an increasing number of sequences due to alignment uncertainty. Finally, technical difficulties hamper the deployment of such genomic analysis workflows - especially in a reproducible manner - often presenting a high barrier for even skilled practitioners. This work aims to address this trifecta of problems through a web server for fast homology extension based MSA, two new methods for improved phylogenetic bootstrap supports incorporating alignment uncertainty, a novel alignment procedure that improves large scale alignments termed regressive MSA and finally a workflow framework that enables the deployment of large scale reproducible analyses across clusters and clouds titled Nextflow. Together, this work can be seen to provide both conceptual and technical advances which deliver substantial improvements to existing MSA methods and the resulting inferences.

Resum

Un alineament de seqüència múltiple (MSA) proporciona una descripció de la relació entre seqüències biològiques on les columnes representen una ascendència compartida a través d'un conjunt implicat d'esdeveniments evolutius. La majoria de la investigació en el camp s'ha centrat a millorar la precisió dels alineaments dins del marc d'alineació progressiva i ha permès inferències poderoses, incloent-hi la reconstrucció filogenètica, el modelatge d'homologia i la predicció de malalties. Malgrat això, quan s'aplica als conjunts de dades de genòmica moderns, que sovint comprenen desenes de milers de seqüències, sorgeixen nous reptes en la construcció d'un MSA precís. Aquests problemes es poden generalitzar per formar tres problemes bàsics. En primer lloc, a mesura que augmenta el nombre de seqüències, les metodologies d'alineació progressiva presenten una disminució espectacular de la precisió de l'alineació. A més, per a un conjunt de dades, existeixen molts MSA com a possibles solucions un problema que s'agreuja amb un nombre creixent de seqüències a causa de la incertesa d'alineació. Finalment, les dificultats tècniques obstaculitzen el desplegament d'aquests fluxos de treball d'anàlisi genòmica, especialment de manera reproduïble, sovint presenten una gran barrera per als professionals fins i tot qualificats. Aquest treball té com a objectiu abordar aquesta trifecta de problemes a través d'un servidor web per a l'extensió ràpida d'homologia basada en MSA, dos nous mètodes per a la millora de l'arrencada filogenètica permeten incorporar incertesa d'alineació, un nou procediment d'alineació que millora els alineaments a gran escala anomenat MSA regressivu i, finalment, un marc de flux de treball permet el desplegament d'anàlisis reproduïbles a gran escala a través de clústers i computació al núvol anomenat Nextflow. En conjunt, es pot veure que aquest treball proporciona tant avanços conceptuals com tècniques que proporcionen millores substancials als mètodes MSA existents i les conseqüències resultants.

Keywords

multiple sequence alignment, alignment uncertainty, regressive alignment, sequence analysis, reproducibility, workflows, containers, web servers, cloud computing.

Preface

The document is structured such that the introduction chapter is intended to provide background for the reader who may not be specifically skilled in the art of multiple sequence alignment methods and inferences. Thereafter, each of the five central chapters contains a manuscript. It is here where the reader will find more grounding material for the specific problem addressed in each. To avoid confusion, with the exception of the introduction and discussion chapters, the references, figures and tables for each of the respective chapters are self-contained. The discussion chapter is intended to provide context for the results of each central chapter. For the sake of brevity, the supplementary material from all manuscripts have been excluded from this document; however, they are available online from the respective journal publications.

Contents

1. Chapter 1: Introduction

1.1 Multiple sequence alignment

Our most fundamental insights in biology arise from comparisons. And for good reason. Comparative biology exploits the single most important fact in life-science: that all life is related. Relatedness has persisted as the central idea, woven with ever-changing technologies, through the fabric of our advances. The concept of shared ancestry - what we call homology - is the basis for so much of our biology and in sketching the iconic tree of life, Darwin initially conceptualised homology through anatomical comparisons, those of beaks, bones and barnacles. Importantly, comparative anatomy provided a quantification of the similarity between species. With rulers in hand, numbers could be compiled and calculations performed paving the way for the scientific method machine and its instruments of observation, measurement and hypothesis formulation. Yet quantification of anatomy is not entirely satisfactory. It allows for the formulation of phylogenetic relationships but provides no direct connection to evolution's mechanism. Variation and natural selection as concepts can only explain so much. The raw material of evolution is the molecule and our core representation of these molecules is the sequence.

Sequences have a unique role in bioinformatics and it is worth devoting some words to them. With a sequence, our measurements of similarity are quantified at single molecule resolution. But more importantly, the sequence

representation opened up a toolbox of quantification methods and allowed biology to stand on the shoulders of mathematicians and computer scientists. From simple edit-distances to complex machine learning algorithms, strings of characters lend themselves amenable to computation.

A linear or 'text-like' structure - what we now call sequence - was first proposed for proteins in the 1880s (Curtius 1883). Fisher and Hofmeister both developed independent peptide theories at the turn of the 20th century by comparing and contrasting the chemical and physical properties of different proteins (Hofmeister 1902). The hypothesis that proteins were made up of chains of amino acids with particular meaning was not fully proven until protein sequences were published. The first, gramicidine S in 1947, a five peptide protein (Consden, Gordon, and Martin 1947) was followed shortly after by Sanger and Tuppy with a section of 30 amino acids from the B chain of insulin (Sanger and Tuppy 1951).

Biochemistry had occupied the minds of many scientists in the early 20th century but the catalytic enzymes themselves were seen as intractable compared to the small molecule steroids and vitamins. Proteins were comparatively huge and proved difficult to separate and purify. And so the story of the first sequences is one of method development. Sanger himself had remarked that "of the three main activities involved in scientific research thinking, talking, and doing, I much prefer the last and am probably best at it" (Stretton 2002). Over a number of years, he and others developed

techniques to isolate and selectively remove the labeled N-terminus of amino acid sequences. Starting with insulin - which was available in both high quality and quantity due to its therapeutic value - the first sequences were collected and comparisons made. This era saw a great multi-disciplinary convergence which foreshadowed what we are currently observing with molecular biology and computer science. The post-war physicists had turned their hand to crystallography and started the first comparative studies on three-dimensional atomic arrangements. Starting with ungapped super-imposition of protein fragments focusing on active sites, they began the search for governing principles to relate sequence, structure and function.

The background training of some biochemists had allowed for a connection to genetics. Neel and Pauling demonstrated respectively that Mendel's inheritance patterns could describe sickle-cell anemia and that gel electrophoresis could separate normal and sickle-cell hemoglobin (Neel 1952), (Pauling and Itano 1949). This discovery marks the advent of 'molecular diseases' and provided a hereditary link to the molecular phenotype observed. In 1959, on the centenary after the *Origin of a Species*, Anfinsen released his seminal textbook *The Molecular Basis of Evolution* in which he highlights this convergence of fields stating how "many scientists, working either in protein chemistry or in genetics, or for that matter in relatively unrelated fields, have arrived at long-range research plans that are

similar to my own, down to almost the last detail of experimental planning" (Anfinsen 1959) .

In collecting sequences for today's big data genomics, our measurements are rarely on proteins themselves. As to our most useful proxy, nucleic acids, the heritory role of DNA had been hypothesized since its discovery by Miecher in 1869. Later in his life he conjectured that DNA could be a "molecular text" consisting of a linear sequence of chemical symbols. But it is hard to disentangle the revisionism from the facts here; these statements were by no means core views of the scientific community. The consensus until as late as 1949 was that DNA was a simplistic, repetitive molecule of four nucleotides. Even as advances in X-ray diffraction shined an electromagnetic light on the nucleic bases of DNA, the notion from Astbury and Bell that these bases could "form the long scroll on which is written the pattern of life" was marginalised. The discovery of both the definitive genetic role of DNA and the base pairing rules paid rest to this (Chargaff and Magasanik 1949). In solving the structure of DNA, a mechanism for Darwin's evolution was realised (Watson and Crick 1953). The translational code linking nucleic acids to proteins became the goal with Nirenberg and Matthaei demonstrating the first triplet codon (UUU/phe) which was soon followed by the deciphering of all 64 codons (Nirenberg and Matthaei 1961). The code was practically universal across species (Woese 1961) and in our best abstraction, it was sequences which where mutated and selectively passed on to future generations.

Nucleic acid sequencing itself would not be routine until 1977 (Sanger, Nicklen, and Coulson 1977). For the intervening period, the protein sequencing effort would continue in earnest, a laborious operation carried out by a handful of laboratories around the world. These earliest studies - including pivotal work on the evolution on haemoglobin chains - presented the sequences one after each other, unaligned (Ingram 1961). The first primitive alignments were published with the aptly title of "Chemical Paleogenetics" by Pauling and Zuckerkandl along with work by Margoliash on cytochrome c in the same month. Data was still scarce, but as the techniques for protein purification, Edman degradation and N-terminal sequencing improved, the first sequence databases became available.

Margaret Dayhoff and colleagues assembly of sequences from 1965 titled the *Atlas of Protein Sequence and Structure* consists of 70 proteins, mainly cytochrome c, hemoglobins, and fibrinopeptides from various species (Strasser 2010). During the compilation, new possibilities for the representation of sequences presented themselves. It had been noted there are all manner of ways to organise and sort protein data and left unquestioned, the human eye will find patterns in the tea leaves. Dayhoff took the liberty of changing the common three letter amino acid code, converting it into a single letter code. Crucially, gaps could be introduced to improve the alignment. With this leap and enough data, a statement on the origins of specific residues could be made. This point must be stressed as it

marks a progression from species having a shared ancestry, to sequences having a shared ancestry and finally to individual residues having a shared ancestry.

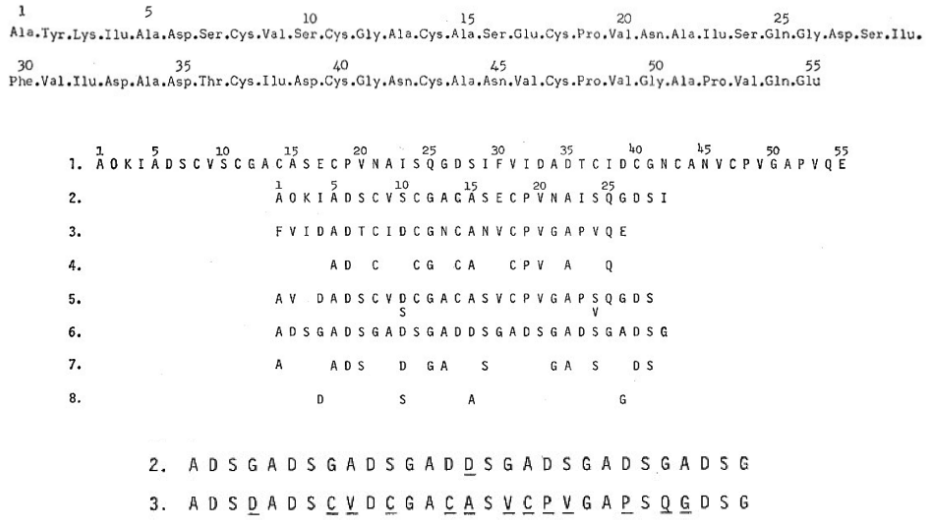


Figure 1: The earliest multiple sequence alignments were created by hand. Adapted from the publication *Evolution of the structure of ferredoxin based on living relics of primitive amino acid sequences* by Eck and Dayhoff in 1966.

The representation of aligned sequences was key. Almost immediately the concept of using this information as a molecular clock arose as formalised in *Molecules as documents of evolutionary history* (Zuckerkandl and Pauling 1965). In aligning residues, a multiple sequence alignment (MSA) is created and a statement implying a series of evolutionary events is made. Gaps represent insertions or deletions and mismatches represent substitutions. From here it was only natural to now quantify this evolution with the help of models.

All models are wrong but some are useful - George E.P. Box.

Herein lies one of the central themes of this dissertation. MSAs can be useful for many things: evolution, structure, function. And yet we are unable to know the truth of a given evolutionary history, hence the absolute correctness of an MSA is practically unknowable. Moreover, we have another layer of inherent uncertainty that arises beyond that of biological intractability. Our representations, scoring schemes and alignment algorithms come along with their own assumptions, something that will be reviewed in more detail in section 1.3 and addressed in the publications that form Chapter 3 and Chapter 4.

With enough data in hand, measurements of similarity could be made. By collecting sequences and aligning them, it was possible to observe the individual substitutions and generate a measure for how often a given substitution occurs. This generates what we now call a scoring scheme. The first scoring schemes were basic and generated from the well studied proteins such as cytochrome c and ferredoxin (McLachlan 1971). At the time of the last edition of *Atlas* published in 1978, the quantity of data allowed for the definition of the first point accepted mutation (PAM) substitution matrices, representing mutations compiled from 1,573 substitutions (Strasser 2010). A PAM considers amino acid changes with silent changes not included (eg AAG to AAA / Lys to Lys is not a PAM). It becomes apparent for obvious reasons that within homologous sequences,

the substitutions of similar physicochemical amino acids are observed more frequently (Glutamic Acid to Aspartic Acid) than dissimilar substitutions (Tryptophan to Alanine). A matrix of PAM-1 is the average observed mutations that is seen when 1% of amino acids are substituted. It is especially important to grasp however that substitution is not a one way street. What we observe are mutations. But those substitutions which have occured twice (reversions or otherwise) may be unseen. Therefore it follows that after 100 PAMs, not all residues have changed. Many positions will have changed and then returned to their original state whilst others will not have changed at all. The PAM-250 matrix was the first published in *Atlas* and is still available for use on the NCBI-BLAST web servers today. Despite some methodological critiques and the vast quantities of new data collected over the last 40 years, the original PAM matrices have proven remarkably robust.

With the advent of larger computationally derived multiple sequence alignment data, the BLOcks SUbstitution Matrix (BLOSUM) series of matrices was developed (Henikoff and Henikoff 1992). These matrices use the log-odd ratio taken from 2,000 blocks of aligned sequences across 500 groups of related proteins and have become the defacto standard scoring matrices for scoring the similarity of protein sequences. As nucleic acid sequencing gained popularity and the significance of non-coding section of a genome was piquing interest, improved scoring schemes for DNA also became available (States, Gish, and Altschul 1991).

It was not lost on the earliest practitioners that with a comparative measure, statistics can be applied to estimate the probability that a substitution has occurred by chance. Rudimentary procedures for searching for homologous sequences had even been developed in the initial sequence comparison work (McLachlan 1971). But to do this in an efficient and objective manner, automated methods for aligning sequences were therefore needed.

In comparing any two sequences, there are generally considered to be two ways to align them. The first is global alignment, where the objective is to match the entire lengths of the sequences to be aligned. Alternatively, local alignment aims to maximise portions of the sequences or substrings. Efficient global alignment made an entrance into biology via Saul Needleman and Christian Wunsch (Needleman and Wunsch 1970). The algorithm that carries their name aims to maximise the similarity between two sequences. The solution they developed can be generalised with the most famous being Levenshtein's approach which seeks to minimise edit distance (Sellers 1974).

In its initial formulation, Needleman-Wunsch begins with a matrix of size len(N) x len(M) with the two sequences to be aligned represented along the edges of the matrix. Once a scoring scheme (including gap penalties) has been decided upon, the first column and row of the matrix are filled additively with the gap penalty scores. Next, and for each cell progressing in

a top-down, left to right fashion, one of three possible outcomes is selected based on the highest score. Taking the score:

- from the top left diagonal, add to it the score from our scoring scheme for the two amino acids those positions representing a match/mismatch in our aligned sequences.
- from the cell above, add to it the cost of a gap, representing the addition of a gap into sequence M of our alignment.
- from the cell to the left, add to it the cost of a gap, representing the addition of a gap into sequence N of our alignment.

The highest score of the three is then entered into the cell. The choice of gap penalty here has an obvious impact on the resulting alignment but also for the implementation (and thus complexity) of the algorithm. To better approximate biological processes, an affine gap penalty concept was developed where the penalty is composed of an initial gap opening cost plus a gap extension penalty.

It is important to note that it is possible for more than one of the three outcomes (match/mismatch, gap in sequence M or gap in sequence N) to have the same score. This becomes important for the second part of the algorithm known as the traceback. Once the matrix has been completed, we start from the bottom right and traceback to the top left following the path to generate the optimal scoring alignment. Given that there can be many optimal paths, there are many optimal alignments.

The key concept of Needleman-Wunsch is that it efficiently removes from consideration those comparisons that are unable to contribute to the maximum scoring alignment. Many improvements were subsequently made to the algorithm in terms of speed and memory usage. A recursive approach was added whilst David Sankoff developed an approach that completes the table in quadratic time (Sankoff 1972). It was determined that storing of the complete matrix was not necessary given that the optimal score only comes from the line directly above or the cell to the left which significantly reduced the memory requirements. Other heuristic improvements included 'banding' of the matrix to eliminate the need for full computation of the matrix at the expense of possibly excluding the optimal solution. In cases where the global alignment is of key importance, these derivatives of Needleman-Wunsch are still in use today.

The consequences of dynamic programming on the newly formed field of bioinformatics proved to be far-reaching. Further modifications of Needleman-Wunsch made the framework applicable beyond that of obtaining the maximum global score of two sequences. When sequences have changed over evolutionary time, meaningful comparisons of the complete (global) sequences may not be highly informative. Often evolutionary forces are applied to domains and motifs as epitomized in the phrase "Nature is a tinkerer and not an inventor" (Jacob 1977). In local alignment this is accommodated by finding the optimal scoring alignment between subsequences of the original sequences.

This was first described by Smith and Waterman (Smith and Waterman 1981). In filling the matrix, the technical modifications to Needleman-Wunsch are rather simple. The first row and column of the matrix are initialised with scores of zero and in computing the scores, any negatively scoring cells are set to zero. During the traceback procedure, we now start with the highest scoring cell and work back to generate the optimally scoring local alignment. Many of the modifications used to improve Needleman-Wunsch could also be applied to Smith-Waterman (vice-versa) as shown by the overlap in the algorithmic development. Significant later advances included Gotoh who reduced the complexity from $O(m2n)$ to $O(mn)$, and Myers and Miller who reduced space requirements to be linear, i.e. to the length of one of the sequences (Gotoh 1982; Myers and Miller 1988).

With the efficient alignment between pairs of sequences computationally possible, methods for sequence search were still absent until the 1980s; if only because the databases themselves did not exist. The first genome had been published in 1977 and it was still common for sequences to be copied by hand from literature (Sanger et al. 1977). The same year saw the initial release of the structure based Protein Data Bank (Bernstein et al. 1977) shortly followed by the first nucleotide database, the EMBL Nucleotide Sequence Data Library in 1980 (Baker 2000). The American effort for coordinating sequence resources coalesced at the national level leading to

the formation of the NCBI in 1987. Its first director would be David Lippman. Lippman, along with Pearson, had two years prior released the FASTA alignment software package which included Wilbur and Lipman's previous algorithm for database searching of nucleic acids and protein sequences (Lipman and Pearson 1985; Wilbur and Lipman 1983).

FASTA and its successor BLAST are in essence sequence alignment algorithms with heuristics based on k-mers. In k-mer search, sequences are first split into tuples of size k (termed k-tuples) and the location of matches between the k-tuples in the sequences are recorded. A year earlier, an algorithm to detect all common subsequences of length k had been developed and applied pairwise to RNA sequences up to 5,000 nt in length (Dumas and Ninio 1982). Wilbut and Lipman used this k-mer search routine and then extended it to only consider regions where a certain number of k-tuple matches are found within a window. In these regions, termed significant diagonals, joining between close diagonals occurs and a banded Needleman-Wunsch dynamic programing type alignment is used to generate alignments and corresponding alignment scores.

When searching a database containing hundreds, thousands or even millions of sequences as can now be done with BLAST, the distribution of alignment scores becomes critical to knowing whether a given 'hit' is significant. Assuming that most of the sequences in a given database have a random relationship with any given query, Wilbut and Lipman first removed the

highest scoring alignments - these appear as outliers and are assumed to be related sequences - and plotted the distribution of the random scores. This distribution looks remarkably like a normal distribution, thus following a transformation allows the use of standard P-value hypothesis testing (Wilbur and Lipman 1983). The implementation and statistical basis for rapid detection of sequence homology was realised.

The original algorithm was improved upon by Lipman and Pearson to allow for global and local gapped alignments with FASTP and FASTA respectively (Pearson and Lipman 1988). But it was the 1990 release of BLAST that changed the course of heuristic sequence searches (Altschul et al. 1990). It is hard to overestimate the importance of BLAST in this story. As the most cited publication of 1990s, it has stood the test of time even as the size of sequence databases increased by many orders of magnitude. On a personal level, performing BLAST on NCBI website was my first introduction to bioinformatics and I remember the profound feeling of wizardry at my fingertips with that first sequence search.

In comparing BLAST to FASTA, the first difference comes from the representation of the query sequences. Instead of attempting to compare every k-tuple of the query with every k-tuple in the target database, BLAST first constructs a list of query k-tuples and - importantly - generates a list of similar k-tuples. It then only keeps the k-tuples which would score above a neighbourhood threshold score. The basis for this is that statistically

significant target sequences should contain these high scoring k-tuples derived from the query sequence. This list of similar words can be stored in a tree data structure that allows for efficient searching. The target database is then scanned for matches to the query k-tuples and the position of exact matches recorded called seed locations. For each seed location, an ungapped local alignment between the query and target sequence is performed, extending the alignment in both directions until the total score of the alignment extending from the seed region decreases. These alignments become high-scoring segment pairs (HSP) whose score is evaluated for significance according to the Gumbel extreme value distribution. When more than one HSP is found within the same target sequence, an attempt is made to combine them into a longer alignment. Finally, all alignments whose expected score is lower than a threshold *E* are reported.

This final point on the *E*-value statistic marks another difference between database searching with FASTA and BLAST. FASTA takes the view that all sequences in our target dataset are *a priori* equally likely to be related to the query. The more nuanced approach of BLAST is that the *a priori* chance of being related is proportional to sequence length. Longer sequences are more likely to be multi-domain and therefore the query is more likely to be related to longer sequences than it would be to shorter sequences (Altschul and Gish 1996). This becomes crucial when considering nucleic acid searches where the target database sequences can comprise of whole chromosomes with lengths in the order of hundreds of millions of nucleotides.

Vast database searching is made possible with FASTA and BLAST; however, there are still trade-offs in terms of specificity when compared to optimal dynamic programming algorithms such as Smith-Waterman. This is noticeable when searching for more distantly related sequences. There is an appreciable 'twilight' zone for detection that occurs at approximately 25% sequence identity for proteins and 60% for nucleic acids. To extend this twilight zone, a BLAST iteration based on profiles termed position specific iterative BLAST (PSI-BLAST) was developed. In PSI-BLAST, prior to searching, a profile is first generated from closely related sequences. This profile can then be used to search a database with significantly similar sequences added to the profile and the search performed in an iterative manner. This approach is very successful and had led to the discovery and characterization of diverse sequences which share a common origin (Altschul et al. 1997).

A sequence profile is a generalised extension of the slightly more intuitive position specific scoring matrices (PSSMs). A PSSM at its essence is a reduced representation of a multiple sequence alignment. At each position the frequency of each character is recorded. PSSMs are able to use the wider information content that comes from aligned sequences and evolutionary constraints. These prove to be very useful for pattern matching of motifs for example but they forbid gaps (insertions/deletions) which prohibits use for longer sequence comparisons. Profiles alleviate the

problem of gaps by allowing insertion and deletion states. At each position, these states have a score derived from the frequency than an insertion or deletion is observed at that position in the MSA. These generalized profiles form the basic representation of the query sequence in PSI-BLAST.

Further improvements to profiles were made with the introduction of the hidden Markov model representation (HMM) as applied to profiles (Durbin et al. 1998). Profile-HMMs differ from the profiles discussed above in that they are able to better contextualise the evolutionary signal through the use of probabilistic modeling, e.g. the probability an alanine in the query sequence matches the model given the previous residue was a matched leucine. Three different hidden states are used in profile-HMMs: match states, insert states and delete states. These three states describe the position-specific frequencies of characters as well as the insertions and deletions frequencies for each position in the consensus sequence (Mount 2009).

Applying the algorithmic framework of HMMs allow us to tackle several fundamental sequence alignment problems using different algorithms. The forward algorithm can be used to calculate how likely a given sequence is to be emitted from a model and thus gives an estimate of the likelihood of homology. The Viterbi algorithm gives the most probable path between states given a sequence and thus the returns the optimal alignment score. Finally to generate and train a profile-HMM from an MSA, the

forward-backwards and Baum-Welch expectation maximization algorithms are used. The use of profile-HMMs was pioneered and popularised through the software developed by Sean Eddy with the HMMER package (Eddy 1998) being central to the building of the most widely used protein family database Pfam (Sonnhammer et al. 1998).

The use of probabilistic models also extends to RNA. In an alternative to the primary sequence based profile-HMMs, there are the related stochastic context free grammars, termed covariance models (CMs) for the purposes of RNA homology search (Nawrocki and Eddy 2013). At a basic level, these can be considered as structured-HMMs where the relationships of columns do not run strictly left to right along a consensus sequence . This allows for the modeling of long range interactions and pairwise structures whose nucleotides may 'co-vary' during evolution. These models along with structure annotated MSAs form the basis of the RNA families database Rfam (Nawrocki et al. 2015).

The history of multiple sequence alignment is not a linear progression of ideas and the previous chronological description defies the reality of the developments. MSA methods have been heavily influenced by the concurrent advances in computation and sequencing technologies and ideas have cross-pollinated from many fields. In the homology searching examples above, we see how the MSA is a means-to-an-end with a certain amount of circularity. The detection of homologous sequences can be improved by

including these very sequences into the seed MSA. In phylogeny applications of MSA, the circularity is even more apparent with the majority of methods relying on MSA built with a guide tree which reflects the resulting phylogenetic tree (Lake 1991). This guide tree approach is formalised as the progressive MSA framework which will be discussed in the following section.

To objectively organise the sequencing data, automated methods for MSA generation were required. This was enabled by the adoption of computers as routine instruments in laboratory. But even with computing power becoming commoning, new methods were needed. If we consider our objective is to optimally align all our sequences using a given scoring scheme, the naive approach would be to expand our algorithms for pairwise sequence alignment. The dynamic programming approach uses a two dimensional matrix which can be extended to three dimensions for three sequences or an n-dimensional lattice for n sequences. However, the alignment space expands exponentially with the number of sequences to be aligned as formalised with a computational complexity of $O(length^{Nseqs})$. In practice, determining an optimal MSA is not possible for all but the smallest of sequence sets. This intractability necessitates alternative heuristic approaches.

The most common heuristic approach is the progressive alignment framework which reduces the problem of aligning all sequences to a series

of ordered pairwise alignments ordered according to a pre-estimated guide tree. Original ideas for the progressive alignment method can be traced back as far as Fitch and Yasunobu who in 1975 generated phylogenies from gapped alignments. Interestingly, they noted the close relationship between where gaps are placed and the resulting phylogenetic trees (W. M. Fitch and Yasunobu 1975). Hogeweg and Hesper were the first to provide an algorithmic description of a progressive procedure (Hogeweg and Hesper 1984). The idea is to start with the pairwise alignment of all sequences to generate a similarity matrix (Figure 2). The matrix can be populated with the alignment score as calculated by the dynamic programming procedure and can then be used to estimate a tree. Importantly, this tree provides the order in which the pairwise alignments will occur. The alignment of leaves in this tree (sequences) does not provide any difficulty beyond the standard dynamic programing approach; however, when aligning nodes, we come across situations where we must either align one sequence against already aligned sequences or align two sets of already aligned sequences. This is termed profile alignment. Slightly more sophisticated dynamic programing algorithms that extend Needleman-Wunsch have been developed for profiles where the sequences that run along the matrix edges are replaced with character frequences. When progressing through the guide tree, resolution of the root node yields the final alignment and completes the progressive procedure.

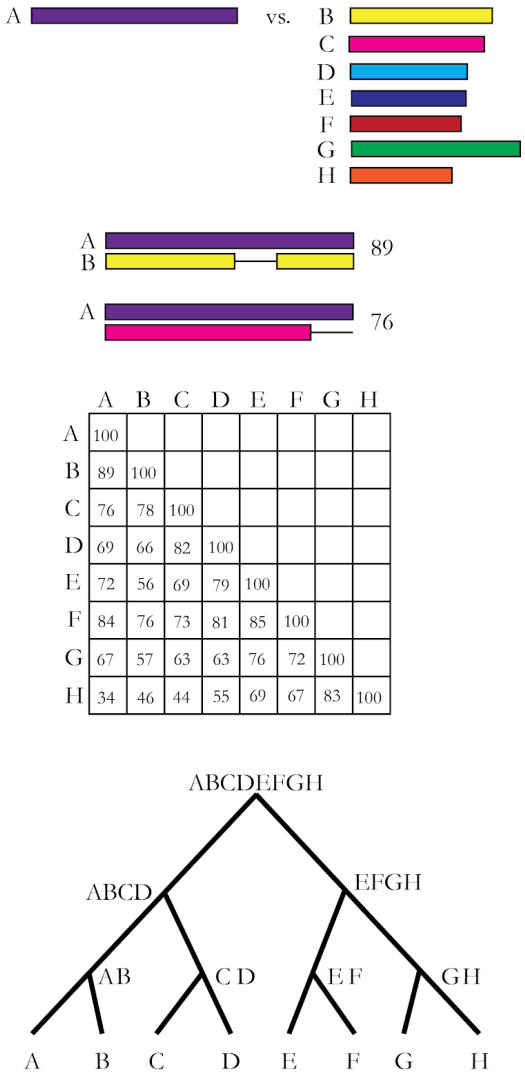


Figure 2: Guide trees. The progressive alignment procedure begins with the pairwise alignment of all sequences. Each of the *n* choose *k* pairwise alignments are scored and the score is recorded into a distance matrix. Given the eight sequences shown here (A to H) , 28 pairwise alignments are performed. This matrix can then be transformed into the guide tree using any distance based agglomerative hierarchical clustering method (bottom). The initial approaches used an average group linkage procedure.

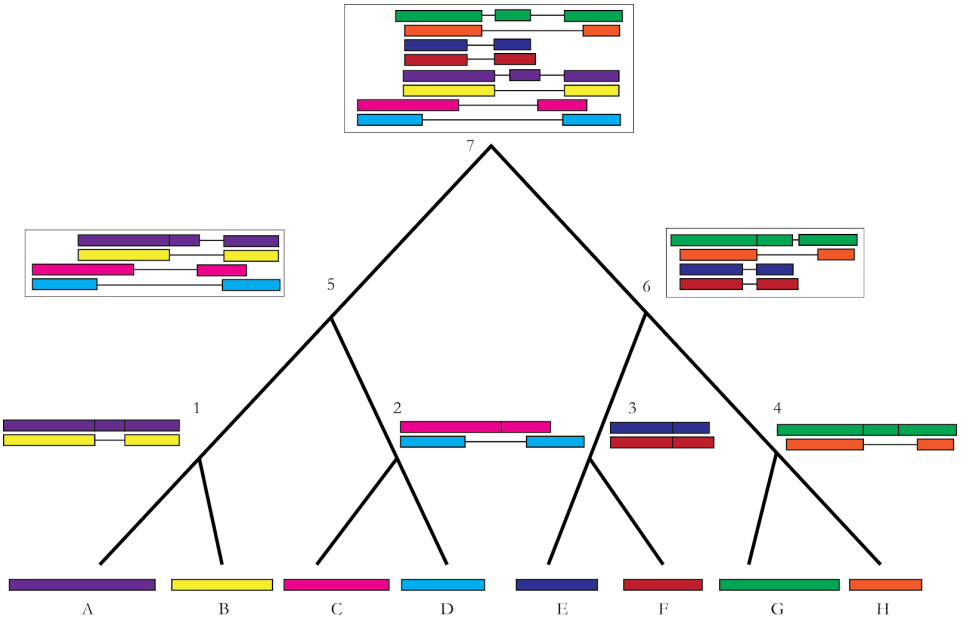


Figure 3: Progressive multiple sequence alignment. Once the guide tree has been calculated, the procedure starts with the pairwise alignment of the leaves. For the eight sequences above, sequence A and sequence B would first be aligned to generate alignment 1. This would continue along the leaves before the profile alignment of alignment 1 and alignment 2 is performed to generate alignment 5. The procedure is complete with the alignment of the final profiles to generate the final alignment at the root of the tree.

Progressive MSA methods have become by far the most widely used approaches for aligning multiple sequences. This observation is reflected in the success of the Clustal software. First released in 1988, the original Clustal (Higgins and Sharp 1988) was a largely faithful implementation of the progressive framework described above. ClustalW has since gone on to become the 10th most highly cited publication in science (Thompson,

Higgins, and Gibson 1994; Van Noorden, Maher, and Nuzzo 2014). It is hard to overstate the effect of ClustalW popularity on the MSA field. Every year there are still hundreds of studies published using ClustalW despite many improvements being available.

One such performance improvement to MSA method involves iterations. In the earliest work, MSA were used almost exclusively to generate phylogenies. Herein lies a circularity in the progressive approach: to generate an interesting alignment, an accurate tree is required and to generate an accurate tree, a good alignment is required. This was not lost in the approach of Hogeweg and Hesper. Once an initial tree is estimated and an alignment created, this alignment can then be used to generate a new re-estimated tree. The procedure can continue on *ad infinitum* in what is termed guide-tree re-estimation. Other iterations involve re-aligning preliminary nodes in the guide tree (Hirosawa et al. 1995). This can be done by separating the preliminary alignment at each node into partitions and performing group-to-group dynamic programming on the partitions (Gotoh 1993). The SP-Score of the alignment resulting from aligning each group (left child and right child) is evaluated with each iteration and those alignments which improve the SP score become child nodes. Iterative approaches are commonly used in many alignment software programs such as MAFFT and MUSCLE (Katoh et al. 2002; Edgar 2004).

Another major improvement was the introduction of consistency-based alignment. In progressive methods, errors made early in the alignment procedure propagate through the alignment due to the "once a gap, always a gap" property. Consistency-based methods attempt to improve this by minimizing these errors through the use of information from different sources beyond the usual global pairwise alignments. The different information sources become libraries, the compatibility of which are used to calculate the consistency score. The most widely used implementation of consistency is T-Coffee (Tree-based Consistency Objective Function for Alignment Evaluation) (Notredame, Higgins, and Heringa 2000).

In the original T-Coffee formulation, both local and global pairwise alignments are performed for all pairs of sequences. Each pairwise alignment can be represented as a list of paired residues with the pairs weighted using sequence identity. The libraries are then combined so that the pairs are weighted according to how consistent the pairs are seen across all the information sources via examining triplets. This extended library is used as a scoring scheme to align all the original sequences in a pairwise manner. The pairwise alignment will therefore better reflect the alignment of residues consistent with all other residues pairs. These pairwise alignments are then used as the starting material for a progressive alignment procedure.

Consistency results in significantly improved alignments. This is particularly true when the identity of sequence is lower. The drawback is the increased

computational resources required over traditional progressive alignment programs such as ClustalW. This effectively limits the approach to the alignment of approximately 1,000 sequences. Consistency-based alignment has since been used in several different algorithms including ProbCons which uses probabilistic consistency-based alignment (Do et al. 2005) and MAFFT-ginisi.

Importantly, with consistency based methods, many different information sources can be used. This has lead to the development of a range of different applications. R-Coffee utilises RNA covariation information as part of the library weighting scheme to accurately align RNA (Wilm, Higgins, and Notredame 2008). Expresso uses 3D structural information to build a library and produces very accurate structural multiple sequence alignments (Armougom et al. 2006). Another flavour is PSI/TM-Coffee which uses profiles built using PSI-BLAST and is presented in Chapter 2.

The previous section provides a general background to multiple sequence alignment. Specific developments that relate to alignment uncertainty are discussed in section 1.3 whilst recent large scale MSA methods are examined in section 1.4.

1.2 Phylogenetic methods

When considering phylogeny and multiple sequence alignment, given the almost ubiquitous prerequisite of an MSA to build an phylogenetic tree, the

interest in an MSA is often consequential. However, it could be considered that they are two different perspectives of the same data. A phylogeny focuses on the relationship between the rows of an MSA (i.e. species or genes), while an MSA refers to the relations between the columns (i.e. residues).

By definition, an MSA performs sequence comparisons. A phylogenetic tree does not have to obey the same limitations. Comparative anatomy originates at least as far back as Aristotle and morphological comparisons were performed throughout the middle ages. However when species were considered as fixed entities through time, there was little room for the idea of shared ancestry. With Lamarck's conceptual leap forward in 1801, a theory and mechanism for how species could change over time was developed. This opened up the possibility for phylogeny as we know it. In the years that followed *On the Origin of Species*, there was a great flurry of interest. Between 1866 and 1867, paleontologists and comparative morphologists reexamined their respective fields through the evolutionary looking glass leading to the construction of the first phylogenies. This was popularised in-part by the work of Ernst Haeckel who coined the term phylogeny (Haeckel 1866).

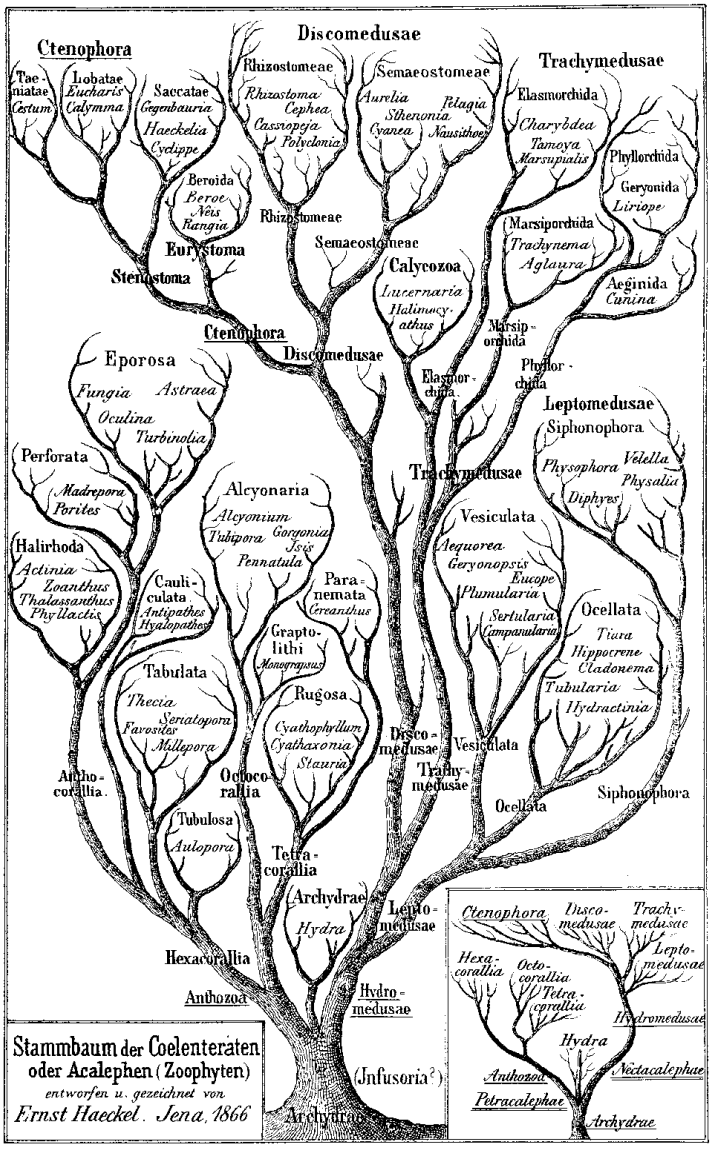
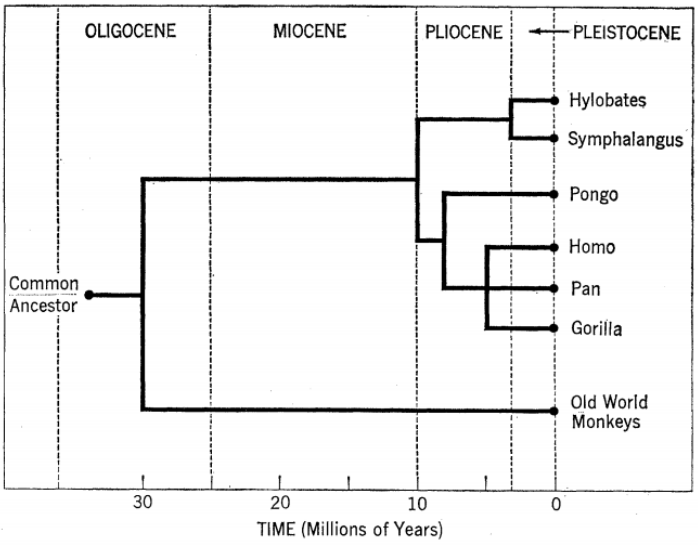


Figure 4: Early phylogenies. Drawing by Ernst Haeckel in 1866. Early phylogenies often conflated species with larger taxonomic groups represented at internal nodes (Haeckel 1866).

Beyond morphology-based measures, and prior to the advent of rapid sequence, molecular comparisons were possible through early molecular studies. The introduction of molecular measures for species comparison provided a far superior proxy for quantifying the underlying evolutionary processes. Zuckerkandl and Pauling coined the term molecular clock after studying the amino acids present in haemoglobins (Zuckerkandl and Pauling 1965). The concept of immunological distance was introduced by Allan Wilson and colleagues using a quantitative micro-complement fixation method. This procedure was used to compare serum albumins across primate species and allowed for the construction of a distance matrix (Sarich and Wilson 1967). The assumption was that these proteins evolve at a steady rate allowing similarity measures to become clockwork. The rate of change observed in the molecules was well suited for the deepest questions relating to hominid evolution and the conclusions drawn have held to be largely true. For example immunological results indicated the last common ancestor between human and chimpanzees to be approximately 5 million years ago. This has been proven to be a far superior estimate compared to the paleontology estimates of between 10 to 30 million years.



Species of albumin	Index of dissimilarity		
	Antiserum to *Homo*	Antiserum to *Pan*	Antiserum to *Hylobates*
Hominoidea (apes and man)			
Homo sapiens (man)	1.0	1.09	1.29
Pan troglodytes (chimpanzee)	1.14	1.00	1.40
Pan paniscus (pygmy chimpanzee)	1.14	1.00	1.40
Gorilla gorilla (gorilla)	1.09	1.17	1.31
Pongo pygmaeus (orang-utan)	1.22	1.24	1.29
Symphalangus syndactylus (siamang)	1.30	1.25	1.07
Hylobates lar (gibbon)	1.28	1.25	1.00
Cercopithecoidea (Old World monkeys)			
Six species (mean ± S.D.)	2.46 ± .16	2.22 ± .27	2.29 ± .10

Figure 5: Early phylogenetic trees of hominids based on immunological distance. The immunological distance matrix (below) is be used to construct a phylogeny. Adapted from Sarich and Wilson 1967 (Sarich and Wilson 1967).

Immunological similarity is simply one measure of molecular distance and has merit for the particular question of primate evolution across this time scale. But to develop metrics that fit more broadly across all of life, ubiquitous sequences are needed. Ribosomal RNA (rRNA) specifically

allows for this quantification. 16S rRNA is essential for protein synthesis and is present in cells from all forms of life. It has sections that are known to evolve at differing rates which provides a multi-scale evolutionary clock. In 1977 Woese and Fox presented the first phylogenetic description linking all three kingdoms of life based on the association coefficient of 16S fragments (Woese and Fox 1977). In the same year, Sanger's rapid DNA sequencing technique was published which soon resulted in phylogenetic analysis using the sequences themselves. With large amounts of single nucleotide resolution data, polymorphism rates could be used to establish more accurate models of evolution (Kreitman 1983). These would subsequently be used incorporated into tree construction techniques that did not rely simply on distance measurements.

We saw in the MSA section how a similarity or distance measures can be used to construct a guide tree. The most simple of these use distance measures, for example morphological measurements or sequence identities. Distance measures are useful in that they allow the construction of trees via pairwise comparisons. UPGMA (Unweighted Pair Group Method with Arithmetic Mean) was one of the first of such methods and results in a rooted tree where the distance from the root to each tip is equal (Sokal, Michener, and University of Kansas 1958). In making these distances equal, UPGMA assumes that the molecular clock - the rate of evolution - is equal across all sequences. For phylogenetic applications specifically, this

assumption makes UPGMA not particularly well-suited for inferring relationships.

In contrast, Neighbour-Joining (NJ) can avoid this assumption whilst still taking a distance matrix as input (Kumar and Filipski 2004). The NJ method begins with a star shaped tree which is decomposed to the final tree in joining interactions. To begin with, the distance matrix is transformed into a Q-matrix which is used for choosing which sequences to join. The sequences with the lowest score (leaf *a* and leaf *b*) in the Q-matrix are joined and a new node representing the ancestor of the now joined sequences is created (node *u*). We next calculate the distance from *u* to *a* and *u* to *b* before updating the distance matrix which now has *a* and *b* removed in place of *u*. This process is repeated until all sequences of the tree are resolved. Importantly, NJ does not require ultrametric data and it results in an unrooted tree where the branch lengths can be interpreted as an approximation for the number of substitutions that have occured.

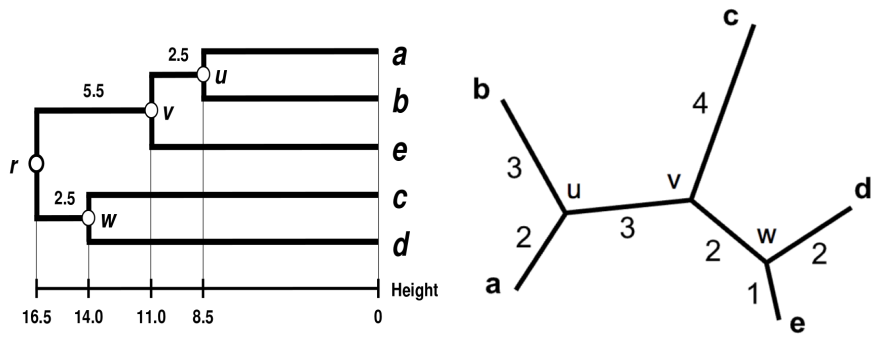


Figure 6: UPGMA and Neighbor Joining trees. UPGMA trees (left) are rooted and have equal branch lengths which assumes the sequence data has a constant-rate of evolution. Neighbor Joining trees (right) are unrooted with branches of differing lengths representing differing evolutionary distances.

The previous methods for tree construction rely on pairwise distances as input. Maximum parsimony on the other hand takes as input characters for each of our taxa (Walter M. Fitch 1971; Farris 1970). These characters could be encoded from categorical data such the presence of absence of an anatomical feature or behavioral trait. For the purposes of this discussion however, our input characters are encoded from a pre-computed multiple sequence alignment. The characters are amino or nucleic acids with an optimality function that selects a phylogenetic tree minimizing the number of character-state changes. It is relatively easy to score the parsimony of any given tree by simply calculating the number of required character state changes. However there is no method to generate an optimal tree given the NP-hard nature of the problem. The number of possible trees in the tree space is huge and it is impractical to find optimal trees in dataset beyond

approximately 10 taxa. There exists heuristic approaches where from an initial sample of possible trees, each tree is scored and the tree with the highest parsimony score selected. This tree is then permuted and the resulting trees scored and selected in an iterative hill-climbing optimisation procedure. The drawback here is that it is possible to get stuck in local optima. Another drawback is that by definition, the most parsimonious tree describes the shorted path explaining the character state changes and not the actual evolutionary history. This results in maximum parsimony methods that underestimate the actual number of evolutionary changes that have occured.

A related approach is maximum likelihood in where the aim is to maximise the probability of observing the data (sequences) given our model (Felsenstein 1981). First introduced by Felsenstein in 1981, the model in this context consists of the tree topology, the branch lengths but also the mathematical description of the process that generated the observed mutations. This description of the evolutionary changes is most commonly modelled as a Markov chain which contains the probabilities of substitutions and the frequencies of the different characters. Analogous to calculating maximum parsimony, calculating the likelihood of any given tree can be performed efficiently using the pruning algorithm. Yet finding the optimal model is difficult and relies on heuristic optimisations. The likelihood landscape is often not smooth due to the discrete nature of different topologies leading to local optima that can be difficult to traverse from. To

search the tree space, "moves" or operations are performed which change the tree topology. The different moves of neighbour interchange, subtree pruning and regrafting and tree bisection and reconnection each have increasing coarse abilities to jump across topological space. Software based on maximum likelihood are among the most commonly used methods for phylogeny construction as popularized through packages including PhyML and RAxML (Guindon et al. 2010; Stamatakis 2014). Extending the principle of maximum likelihood, bayesian methods incorporate a prior probability into the likelihood measure. Here there is an underlying prior probability distribution of possible trees so the Markov chain can be constructed such that it has the desired distribution in a Markov chain Monte Carlo method. Bayesian based approaches have been made available through software such as MrBayes and BEAST (Cummings 2004; Drummond and Rambaut 2007).

The methods described above assume a belief in the probability of the evolutionary model. Alternatively we can use nonparametric methods to assess the reliability of a given phylogeny after we have constructed it. Nonparametric testing is less dependent on the evolutionary model and uses the empirical evidence in the data to assess the robustness of a phylogeny. The first application of nonparametric assessment as applied to phylogentics in 1982 Mueller and Ayala who used 'jackknifing' to determine the validity of UPGMA branch lengths (Mueller and Ayala 1982). The jackknife procedure is a resampling technique that involves systematically removing observations

from a dataset and then re-estimating the phylogeny with the reduced data. The procedure is repeated many times to create a series of replicates. Shortly after the application of the jackknife, Felsenstein applied the bootstrap procedure which has become the defacto non-parametric test of phylogenies (Felsenstein 1985). Bootstrapping is similar to jackknifing however it involves resampling with replacement. This is very amenable to phylogenetics if we consider an observation to be a single column in a MSA. It is possible to resample otheur MSA, creating many replicate MSA of the same length as the original. The bootstrap replicates are able to inform us of the variation that arises from resampling and provides an estimate for the variation in the true but unknown underlying distribution.

Applying the bootstrap procedure to create replicate bootstrap trees is relatively straightforward. Replicate MSA are generated by resampling the original MSA columns with replacement. For a given MSA replicate, some sites may have been sampled multiple times and some sites may not be included. This is done for each replicate with typically 100 replicates generated but dependant on the specific dataset. One assumption for the bootstrap procedure is independent observations. If we consider our characters to be sites, or columns in a MSA, then we must assume that they evolve independently. This assumption is patently incorrect as the evolution of some sites is highly dependent on the context of other sites. However for practical purposes, Felsenstein's bootstrap has proven to be a reasonable

first approximation of the true confidence of the clades (Efron, Halloran, and Holmes 1996).

Given a collection of replicate trees, an estimation of some parameter is required to assess the correctness. With continuously distributed parameters, for example a mean, it would be possible to plot the distribution of parameters from our resampled collection and get an estimation of variance. Yet tree topologies are discrete. One solution is to create a majority-rule consensus tree. This method predates the bootstrap and was introduced in 1981 by Margush and McMorris (Margush and McMorris 1981). A consensus tree is created by first quantifying how many times we observe a given split (partition) in the replicate trees. The partitions that occur in the majority of replicates are retained resulting in a final consensus tree. We can extend this by observing the proportion of times we see a given partition in the replicates and allows the quantification of support or lack-thereof for any partition.

This concept of the support values combined with the concept of alignment uncertainty makes up the methods contained within manuscripts of Chapters 3 and 4.

1.3 Alignment uncertainty

In previous sections we introduced MSA and an important downstream inference, phylogeny. An MSA allows us to make inferences on several evolutionary parameters beyond that of phylogenetic trees. An MSA is often a single step in a pipeline of processes that make up any given genomic analysis. And yet, almost without exception, the MSA is taken as a single observation of truth. The inference from an MSA is based on the observed molecular characters whilst often taking the homology of those characters, i.e. the column structure as a given.

Before discussing alignment uncertainty, it is important to clarify the scope. Alignments usually have a meaning dependant on the context of their application. For example, in a phylogeny study, residues sharing the same column are implied to have a strict common ancestry. For a molecular biologist studying enzymatic roles, the residues may reflect function. A protein structure study on the other hand may wish to align residues based on 3D superimposition. For the purposes of this discussion, the truth is assumed to be phylogenetic.

The weakness of relying on a single MSA is a well know but a somewhat ignored issue. With reference to the progressive alignment framework, in 1991 Lake detailed how the guide-tree has a major impact on the maximum parsimony tree inferred from the MSA (Lake 1991). The order of the alignment, as defined by the guide-tree structure, becomes reflected in the

tree topology. Further the tunable parameters of the alignment method act as a proxy for the underlying evolutionary process we are attempting to deconstruct. The gap-opening penalty should correspond to the indel-rate, the gap extension penalty to the average indel length whilst the mismatch penalty should reflect the percent identity of the sequences we are aligning. The guide-tree itself is a distance-based phylogenetic parameter. This creates a circular dependency in that these parameters are best estimated with a correct MSA in hand. For the most part, a high quality phylogeny estimation requires a high quality MSA whilst a high quality MSA requires a high quality estimation of phylogeny.

There have been attempts to overcome this circularity through joint estimation of the mutually dependant alignment and inferred parameters. Thorne et. al. include the insertion-deletion and amino acid replacement rates of all pairwise alignments in an attempt to capture the regional heterogeneity of replacement rates (Thorne, Kishino, and Felsenstein 1991). However the most common approach to study the robustness of our inferences results from exploration of the parameter space. Morrison and Ellis examined the effect of the gap-opening penalty and gap-extension penalty on the resulting phylogeny from neighbour joining, parsimony and maximum likelihood methods. The aim was to determine how different alignments affect the resulting phylogenetic trees. In their case study of apicomplexa 18S rDNAs they concluded that "different alignments produced trees that were on average more dissimilar from each other than

did the different tree-building methods used" (Morrison and Ellis 1997). This result was attributed to taxa towards the tips of a tree being sensitive to the MSA method. These taxa are more sensitive to the MSA method than to the tree-building method. In exploring the gap-opening and gap extension steps they conclude, at least with respect to ClustalW, that these parameters govern the tension of the alignment (gappiness) yet do not have a major effect on the resulting phylogenetic trees.

When considering the alternative alignments resulting from parameter space exploration, it was observed that some regions were inherently more ambiguous than other regions. One such early study generated 15 alternative alignments by differing the gap penalties. The alternative alignments could then be used to score columns based on their observed frequency in the alternative alignments (Gatesy, DeSalle, and Wheeler 1993). The first applications of this method on inferences was to simply remove or 'cull' ambiguous columns prior phylogenetic tree building. This concept of removal of regions had previously been performed manually in a subjective manner based on the gappiness observed. The exploration the parameter space however provides an objective criteria for uncertainty. Other culling or trimming methods such trimAl and G-blocks are still popular today but do not rely on alternative alignments to provide an assessment of robustness (Capella-Gutiérrez, Silla-Martínez, and Gabaldón 2009; Castresana 2000).

As an alternative to removal of ambiguous regions, Wheeler et. al. developed a method to incorporate alternative alignments into single alignment to be used for downstream inferences (Wheeler, Gatesy, and DeSalle 1995). It is based on the idea of eliding (to join together or merge) and up-weights the signals in common columns whilst down-weighting the signal in variable columns. The source alternative alignments are first generated by varying the alignment parameters as discussed previous. The authors concluded that culling ambiguous regions results in robust trees, but which are conservative with many unresolved taxa. By including the ambiguous regions of an alignment and weighting them in an appropriate way, more accurate inferences can be made.

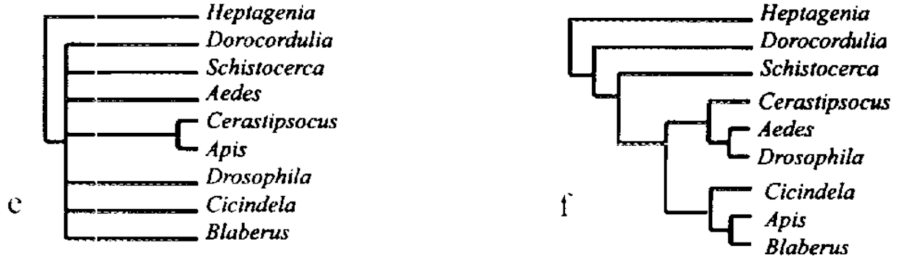


Figure 7: Phylogenies resulting from culled alignment versus elided alignments. Phylogenies from trimmed or culled alignments (left) are robust but often unresolved. Methods such as Elison (right) result in trees attempt to alleviate this by including all alternative alignments and weighting columns accordingly. Adapted from (Wheeler, Gatesy, and DeSalle 1995).

The previous examples consider the parameter space by examining alternative alignments from a single aligner. However we can also consider

the aligner as black-box and perform inferences from it. Alignment uncertainty more recently brought into the scientific mainstream in 2008 with a publication in Science by Wong et. al. in which different alignment methods were used to generate MSA upon which phylogeny and positive selection rates where estimated (Wong, Suchard, and Huelsenbeck 2008). An important point stressed in the publication was that in era of comparative genomics, evolutionary processes are inferred across thousands of genes and taxa. The assumptions applied when considering carefully selected genes on curated datasets become impractical for large datasets. Using a collection of 1,502 orthologous genes across seven species of yeast, the authors show how different MSA methods can result in very different inferences. When considering the tree-topology of maximum likelihood trees, seven of the most popular MSA methods produced alignments that resulted in different trees on 46.2% of the 1,502 gene sets. Inferences of synonymous versus nonsynonymous substitution rates were also shown to be sensitive to alignment method with 14.8% of sites differing in classification at 0.05 false positive threshold. Another important result showed how the bootstrap support for the phylogenetic trees correlates with variability in the alternative alignments. In cases where bootstrap values were shown to be low for a given tree, the alignments were generally more dissimilar.

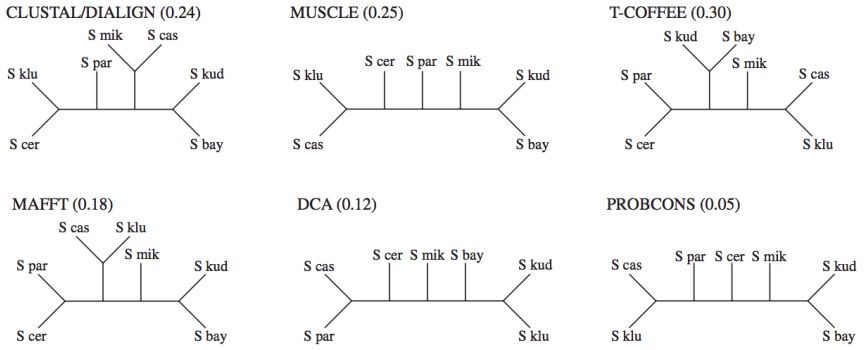


Figure 8: Different alignment methods can result in different inferences. In this example adapted from Wong et. al. 2008, the yeast gene YPL077C alignment produced from seven different alignment methods produce six different estimated trees (Wong, Suchard, and Huelsenbeck 2008).

The combining of multiple MSA methods was described in 2006 with M-Coffee which generates MSA from several methods and then uses consistency to generate a final alignment (Wallace et al. 2006). M-Coffee was shown to outperform the individual methods themselves on the HOMSTRAD, Prefab and Balibase benchmark sets. This work led into methods for evaluating alignments accuracy based on an extension of consistency scoring termed the Transitive Consistency Score (TCS) (Chang, Di Tommaso, and Notredame 2014). TCS adopts the CORE method which uses consistency to score but normalizes this after considering the maximum possible of all possible pair combinations (Andrade 2003). Whilst not relying on alternative alignments, the TCS score is independent from the library generation, so any source of pairwise alignment can be used to populate the pairwise library. Further, TCS provides three different scores which are

applied to the residue, column or alignment as a whole. Using the column score it is possible to provide a lossless alternative to the column removal methods described earlier which results in significantly better estimates of structural accuracy and more accurate phylogenetic trees.

The TCS method is important in that it does not rely on multiple MSAs. However it is difficult to decompose the sources of uncertainty. To do this we must examine the underlying origins of alternative alignments. One such source is co-optimal solutions. With the dynamic programming algorithms described in the first section of this chapter, we saw how it is possible for there to be more than one pairwise alignment with the same optimal score. The choice of alignment is arbitrary and often hard-coded into the algorithm implementation. However when considering the progressive framework with the "once a gap, always a gap" principle, we can see how small errors early in the alignment procedure can result in large discrepancies in the final alignments. To date, examining co-optimal solutions within MSA has been most elegantly handled with a method termed Heads or Tails (HoT) (Landan and Graur 2007). The first version of the algorithm performed the MSA once with the input sequence as is, and then simply reversed each of the sequences and ran the MSA procedure again. The concept is that sequences which were on the horizontal axis of the dynamic programming matrix in the first instance, get placed on vertical axis of the matrix in the second alignment procedure. This examination of the "high-road" and "low-road" provides a way to quantify which regions of the alignments are

sensitive to co-optimal solutions. The method was later expanded upon to perform the reversal of the sequences at each step alignment (internal node) as governed by the guide-tree. This step explores more of the co-optimal space but is limited in application. It becomes a non-trivial problem to re-engineer each MSA software package to perform such operations. Indeed some MSA programs such as PRANK now have options to have the selection of co-optimal alignments performed randomly (Löytynoja 2014).

Another important parameter in the alignment is the guide-tree as illustrated by Lake in 1991. If we consider the guide tree to simply be a distance-based phylogenetic view of our dataset, we can apply the same nonparametric support tests that are traditionally applied to phylogenetic trees. It is this concept that was applied in Guidance (Penn et al. 2010). Uncertainties in the guide tree can be quantified through bootstrap methods to generate bootstrap replicates of the guide trees. These guide-tree replicates can be used to generate alternative MSA, with one alternative MSA for each replicate guide-tree. The Guidance score for each column of an alignment can be evaluated based on the frequency it is observed in the collection of replicate MSA.

More recently Guidance 2 was introduced which provides a measure of uncertainty by combining HoT and the original Guidance as well as gap opening and extension parameter exploration (Sela et al. 2015). These methods provide value in that they provide alternative alignments and have

the ability to identify the source of the alignment uncertainty. However, they are not flexible in that they require significant work arounds for each alignment method. To date, Guidance 2 is configured to run with ClustalW, MAFFT and PRANK alignment methods. Alternatively the TCS score is independent of the methods used. It is however limited to a size of approximately 1,000 sequences due to the computational complexity of the consistency framework.

In Chapter 4 we describe a procedure that can be easily applied to the majority of MSA methods to assess alignment uncertainty and is able to handle datasets with many thousands of sequences that are aligned with large scale MSA methods.

1.4 Large scale MSA methods

We saw in the first section of this chapter how the NP-complete nature of the MSA problem required heuristic approaches, the majority of which have been built around the progressive alignment framework. Historically, we can consider these algorithms to be roughly split into two categories, fast and and accurate.

The most accurate of these heuristic approaches implement some form of consistency-based algorithm. The strength of consistency is in its ability to evaluate all pairwise matches taking into account their compatibility with all other pairwise alignments. Yet this procedure comes at a significant cost. In

practical applications, consistency-based methods such as T-Coffee are unable to align more than a few hundred sequences with MAFFT-L-INS-i being limited to approximately 200 sequences with a maximum length of 2,000 characters.

The fast methods including the original ClustalW, MAFFT and MUSCLE software packages avoid this limitation at the expense of accuracy. However there are still some limitations on these methods which arise from the initial guide-tree construction. The distance-based tree-building procedures which dictate the order in which the sequences are aligned (NJ and UPGMA) have computational complexities of between $O(N^2)$ and $O(N^3)$ depending on the implementation. These approaches become practically impossible beyond a few thousand sequences with 100,000 sequences requiring the computation of approximately 5 billion distances to generate a guide tree. For this reason, the first major innovation required for datasets above several thousand sequences has focused on speeding up this step. The guide-tree problem can be generalised as an agglomerative hierarchical clustering problem. Originally the distances upon which the clustering was performed were based on the accurate but slow Needleman-Wunsch algorithm. The next generation of aligners moved to word-based distance measures to speed up the comparison, however this does not alleviate the required quadratic time of the all-versus-all comparisons and subsequent tree construction.

The most obvious way to reduce the time and memory requirements of tree building is to reduce the number of comparisons performed. This was first successfully applied with the PartTree algorithm where a subset of the sequences are selected and clustered recursively (Katoh and Toh 2007). Beginning at top and then at each level of the recursion, the longest sequence, the sequence with the lowest similarity to the longest and n - 2 random sequences are selected where n is the group size defined by the user. These seed sequences are then used to construct a UPGMA tree and each of the remaining non-seed sequences are associated to one of the seed sequences to create a new group. The same procedure is performed on each of these groups recursively until all sequences are at the leaf of a tree. The final tree can be constructed from the expanded trees. This results in dramatic speed up and reduces the time complexity to quasilinear $O(N \log N)$. In practical terms the authors show that PartTree can align ~60,000 sequences in a matter of minutes using standard desktop computing hardware. PartTree is implemented as part of the MAFFT software package in which it shows a slight decrease in accuracy compared to full tree building methods when benchmarked with Pfam.

Another related method that avoids full distance matrix calculation is the mBed algorithm (Blackshields et al. 2010). Like PartTree, in mBed we first select a set of seed sequences but based on a constant stride selection from the length sorted total dataset. From these seed sequences, reference points can either be refined or not, but in either case the distance between every

sequence and the reference points are calculated. These distances then become a vector for each sequence which contain the coordinates from that sequence to the reference points. The vectors are approximations for the distance between sequences such that we can create an embedded distance matrix which can be used to create guide trees using UPGMA. For very large dataset (over 100,000 sequences) there exists the possibility to use k-means clustering to cluster the vectors directly without the need for an embedded distance matrix. mBed is implemented as part of the Clustal Omega software package (Sievers and Higgins 2013). When evaluated on the the 10 largest Pfam/HOMSTAD datasets, the mBed method took less than 7% of the time that is used to construct the full distance matrix with an average of difference of alignment accuracy of 1.9%. Beyond the optimisations in guide-tree construction, Clustal Omega also utilises an HMM aligner. This differs from the standard profile-profile dynamic programming approach in that it aligns profile-HMMs using HHalign which has been shown to result in more accurate alignments.

Another large-scale method UPP uses HMMs in slightly different manner (Nguyen et al. 2015). UPP first randomly selects a subset of sequences from the dataset and generates a backbone alignment and guide-tree. From this clustering of the backbone sequences, an ensemble of HMMs are built using the HMMER software package. The original sequences which were not part of the backbone are then aligned to the HMMs and the best scoring

incorporated into the alignment. These ensemble HMMs are analogous to the seed sequences described in the methods above.

One final method which employs a very similar strategy is MAFFT sparsecore (Yamada, Tomii, and Katoh 2016). In this procedure, the sequences are first sorted by length before a random selection of 500 sequences are taken from of longest 50% of sequences. These become the core sequences. An MSA is created from this core using the accurate G-INS-i method before the remaining sequences are added to the core using a progressive alignment method.

We see in a number of recent applications in large scale MSA methods a trend towards separating the heuristic agglomerative hierarchical clustering step from traditional progressive alignment step. This separation of clustering and alignment methods forms much of the inspiration for the regressive alignment procedure which is described in Chapter 5. The approach described is generalised so it can combine any clustering method with any alignment method to produce efficient and accurate alignments of tens of thousands of sequences.

In applying these methods to large scale datasets, significant challenges present themselves. The requirements of computationally intensive analysis can include the complex orchestration and management of tasks. This is especially true when trying to adhere to principles of reproducibility. The

analysis described in Chapter 3 provided an initial motivation to explore methods for developing and deploying such large scale MSA analyses. This resulted in the Nextflow workflow platform described in Chapter 6.

1.5 Reproducible workflows and deployment

Retrospectively, the concept of a workflow has existed since the advent of the scientific method. Indeed a methods section is simply a description of actions that should guarantee the repeatability of a given experiment. We can likewise define a workflow to be an orchestrated and repeatable sequence of actions that transform inputs into desired outputs. With the adoption of computing into scientific fields, the use of workflows and computational pipelines has became integral. Today we think of workflows as the combination of different software packages to perform a series of operations on data. Yet any given piece of software could internally be considered as a workflow, and likewise a workflow can considered to be a piece of software in its own right. For the purposes of this introduction I will narrow our definition of a workflow to a description of software steps to perform a genomic analysis.

The humble Bash script has long been used by bioinformatics practitioners. As the most commonly used Unix shell, Bash provides a collection of useful features which make it amenable to writing workflows. Filename globbing allows wildcard matching of input files, the piping between steps allows of processes to be chained together, variables can be defined and methods exist

for conditional testing and iterations. In common usage, a Bash script provides a simple top-to-bottom description of the command line operations that could be typed into the text-based shell terminal. Whilst powerful for simple tasks, Bash scripts are error prone and are not designed to handle complex parallel and distributed computation as modern real-world computational pipelines often require.

An alternative to Bash scripting is GNU Make which was originally developed to automate the various compilation phases required to build software packages in an executable format. It uses the concept of targets which define the desired output of the steps. The targets are contained in rules which specify recipes explaining the actions (commands) to perform on the files to produce the targets. The main concept here is the dependency and relationships between the targets which allows for a bottom-up definition of the workflow beginning with target files. Make has advantages in that the tasks can be implicitly parrelised based on the dependency graph. The correct re-execution of tasks also becomes implicit based on changes in the targets.

Both Bash and Make require a certain level of technical ability to develop the types of workflow routinely deployed in genomic analysis. Data science skills are increasingly valued across all disciplines however the vast majority of university graduates today still lack even basic data handling skills beyond point and click spreadsheets. Graduates trained in biology have traditionally

been the main developers of their own analysis workflows and as such tools which accommodate this level of technical proficiency are popular. The most successful platform of this ilk being Galaxy (Afgan et al. 2016). Galaxy allows biologists with little programming experience to conduct computational analysis through a graphical user interface in a web browser. It relies on either a publically or locally installed server mained by an administrator. Workflows are defined using a drag and drop type functionality. There exists a large collection of tools in wrappers which allow commonly used software to be integrated efficiently into the workflow (Blankenberg et al. 2014). Likewise data services allow inputs to be remotely referenced and sourced. As well as providing a method to define a workflow, Galaxy provides a back-end engine which allows execution with queuing systems commonly used in HPC systems. This distinction here between the workflow definition and the workflow engine is important as highlighted by the common workflow language (CWL) initiative.

CWL is a specification for the definition of workflow applications in a portable manner not only across hardware environments, but also across different workflow engines (runners) implementations. It is yet be seen how this top-down approach of defining a specification and then having the community develop the software will play out. In practice, the vast majority of users rely on the reference cwl-runner engine. Alternatively there are commercial implementations such as the Seven Bridges Genomics platform (Malhotra et al. 2017) or Arvados by Curoverse. The CWL specification is

detailed but presents users with a significant obstacle. Complaints often include the verbosity of even simple pipelines. Likewise, it is not clear how this interoperability of workflow runners ensures reproducibility at the level of the workflow as a whole. By definition, different engines are different implementations written with different code which raises obvious questions in reproducing workflow logic. Minor variances in underlying libraries used by rounding and sorting functions have the potential to obliterate the reproducibility characteristic of portable workflows run on different implementations.

An alternative to CWL is the Workflow Definition Language (WDL) developed by the Broad Institute. Like CWL, WDL is a workflow language specification, however it has been designed in-house by the same team focused on their own workflow management engine termed Cromwell. The tag-line for WDL is very telling in reference to CWL: "Finally a workflow language meant to be read and written by humans". There are efforts to bring two together with planned support for CWL planned for Cromwell 30 onwards. More significantly, initiatives such as the Global Alliance for Global Health (G4GH) attempt to provide broader harmony between workflow engines and languages by providing further specifications. For example the task execution schema is an effort to define a standardized schema and API for describing batch execution tasks whilst the workflow execution schema is a common API which describes how to submit a workflow to a workflow execution system. One characteristic of WDL is the

inclusion of the runtime specification in the task definition. This goes against principle of separating workflow logic from the runtime engine. With respect to workflow portability, this is best achieved through separation of workflow logic from the runtime.

Another popular tool is Snakemake which is based on the Make philosophy described above (Köster and Rahmann 2012). Snakemake consists of a language which is made of rules similar to Make but written in python. It extends the python programing language to be a domain specific language. Snakemake is also has an engine for running Snakefiles and is considered lightweight and portable. It can run on HPC and cloud environments with Kubernetes support.

The previous workflow languages and/or engines provide a way to define the overall workflow logic. They commonly split the operation into execution steps which can be run either locally, sent as jobs to a HPC queue or spun up as an instance in the cloud. For this model to work efficiently, there is a requirement that the tools are available at the place where the computation occurs. The packaging of tools has recently been revolutionised by containerisation technology. The ability to isolate the execution of software tools was initiated by virtual machines (VMs) which are an emulation of a complete computer system. Every VM runs a virtual copy of all the entire hardware an operating system requires to run. VMs are very useful for some tasks however they use up a lot of resources and are slow to

initiate which makes them not well suited for running thousand of small jobs as routinely happens in genomics analyses. Containers provide an alternative as popularised through the adoption of Docker and more recently Singularity (Boettiger 2015; Kurtzer, Sochat, and Bauer 2017). These technologies differ from VMs in that they do not provide any hardware virtualisation, moreover containerised software share the operating system kernel with hosting environment. By using a layered file system and the host kernel, only the required container processes are run which reduces the container overhead dramatically. When applied to typical genomics pipelines, Docker containers have been shown to have little effect on the required resources (Di Tommaso et al. 2015). This portable approach has consequences in terms of reproducibility whilst also facilitating the transition that is occuring with moving the compute to the data and not vice-versa (Pulverer 2015).

One further consideration for workflows platforms is collaboration and sharing. Both software development and science are increasingly collaborative endeavours often conducted across the world between people who will never physically met. The concept of social coding has become popularised by platforms such a GitHub and GitLab which provide users the opportunity to publish, review, and discuss software.

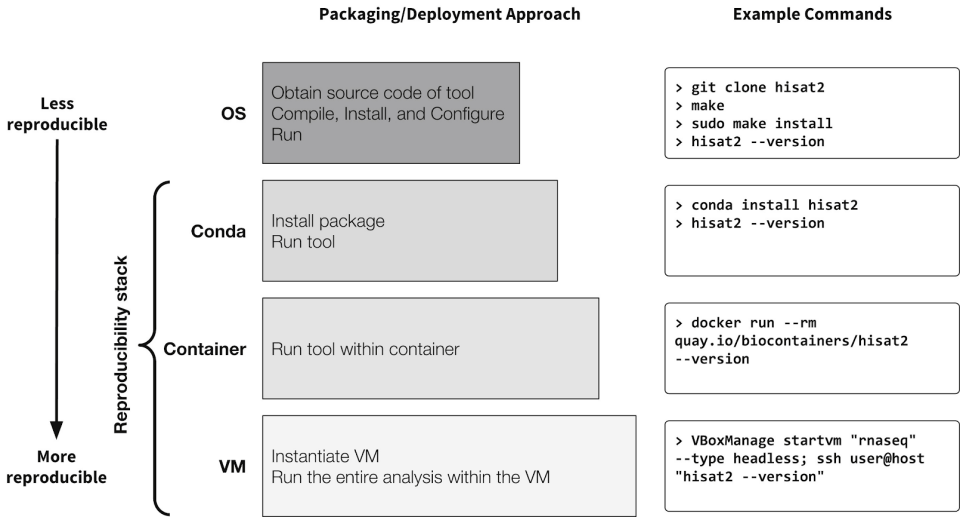


Figure 9: Different levels of reproducibility. In this example of deploying the hisat2 mapping software, different reproducibility stacks are shown. Adapted from (Grüning et al. 2018).

Chapter 6 describes the Nextflow workflow platform and highlights the real problem of reproducibility in genomic workflows which can be solved through containerisation, code sharing and portable deployment.

Chapter 2

PSI/TM-Coffee: a web server for fast and accurate multiple sequence alignments of regular and transmembrane proteins using homology extension on reduced databases.

Floden EW, Di Tommaso P, Chatzou M, Magis C, Notredame C, Chang JM.

. Nucleic Acids Res. 2016 Jul 8;44(W1):W339–43.

Chapter 3

Using alignment uncertainty to improve phylogenetic bootstrap reliability

---

Evan W. Floden, Javier Herrero, Olivier Gascuel, Paolo Di Tommaso, Cedric Notredame, Jia-Ming Chang,

. Bioinformatics. 2019 Feb 6. DOI: 10.1093/bioinformatics/btz082

Chapter 4

Generalized bootstrap supports for phylogenetic analyses of protein sequences incorporating alignment uncertainty

---

Chatzou M, Floden EW, Di Tommaso P, Gascuel O, Notredame C.

Syst Biol. 2018;67(6):997–1009. DOI: 10.1093/sysbio/syx096

Chapter 5

Regressive computation of large scale multiple sequence alignments

---

Edgar Garriga, Paolo Di Tommaso, Cedrik Magis, Ionas Erb, Hafid Laayouni, Fyodor Kondrashov, Evan Floden*, Cedric Notredame*

* co-corresponding authors

To be published

Multiple sequence alignment is one of the most commonly used modeling technique in biology (Van Noorden, Maher, and Nuzzo 2014). MSA models are routinely used for evolutionary and structural reconstruction as well as function prediction. Given the most common scoring functions, computing an exact MSA is an NP-Complete problem that can only be approximately solved using a heuristic approach like the one implemented in the progressive algorithm. This algorithm is at the core of most aligners. It is an agglomerative procedure requiring a pre-computed guide tree that used to incorporate sequences one by one, starting from the leaf up to the root. At every node, a pairwise dynamic programming procedure merges sequences (leaves) or intermediate MSAs treated as profiles (internal nodes). We show here how the same guide trees can be used to incorporate the sequences in the opposite order, starting from the root all the way down to the leaves. This approach that we named 'regressive alignment' yields significant benefits both in terms of scalability and accuracy.

Limitations in the scaling up of the progressive algorithm were initially uncovered in the Clustal Omega benchfam analysis (Sievers and Higgins 2013). Until then small scale empirical analysis had supported the expectation that increasing the number of sequences in an MSA would lead to more accurate models (Katoh 2002). In the ClustalO benchmarks, reference sequences with known 3D structures were embedded in very large datasets (up to 93,681 homologues) and their projected alignments were compared with independently derived structure based reference alignments.

Against all expectations, sequences embedded with more than a thousand homologues proved to be less accurately aligned than when aligned on their own. The effect worsens when increasing the number of homologues. Three recent attempts were made to address this problem, the first one, the chained algorithm (Boyce, Sievers, and Higgins 2014) depends on a processive tree in which every node has at least one leaf child, it brings modest improvements but was heavily criticized (Yamada, Tomii, and Katoh 2016) for its reliance on unrealistic biological assumptions (Tan et al. 2015). The two most recent alternative, UPP (Nguyen et al. 2015) and MAFFT-sparsecore (Yamada, Tomii, and Katoh 2016) rely on a similar principle that involves selecting a subset of representative sequences, turning them into an HMM which is subsequently used to incorporate all the sequences in the final model.

These three approaches all share a similar component: the seeding of the computation with a smaller MSA and the controlled incorporation of the remaining sequences. The main difference is in the selection of sequences than form part of the seed MSA. This approach sets all these methods significantly appart from a regular progressive approach where balanced internal nodes usually leads to the pairwise alignment of large sub-MSAs. We worked under the hypothesis that site degeneration is the main source accuracy loss when scaling up. Based on this, we designed a regressive algorithm meant to generalize the seeded MSA approach by fulfilling two simple conditions: aligning small datasets and not relying on profile

alignments. A procedure consistent with these two constraints can be implemented using a recursive clustering approach that given *M* sequences produces *N* smaller non overlapping sub-datasets each contributing a representative sequence. The MSA of these *N* representatives is the first parent MSA. During the next iteration, the same algorithm is applied onto each sub-dataset and collects a maximum of *N* new representative sequences - with the extra constraint of including the original representative of the whole subgroup within the representative set. Each of the *N* sequence within the first parent therefore occurs both in this parent MSA and in one of its *N* children MSA. This procedure is carried out recursively until each sequences has been incorporated in at least one MSA, thus yielding a maximum of *M*/*N* MSAs, each containing a maximum of *N* sequences.

Since each parent MSA shares one sequence - the representative - with each of its children MSAs, these MSAs can be efficiently merged (Figure 1a) without any need for profile-profile alignments. This is done by treating each residue in the common sequence as a connector between the two corresponding child and parent columns. Insertions occuring in the child MSA are projected in the parent as deletions (i.e. insertion of a block of gaps within the parent) while insertions occuring within the parent guide sequence are treated in a symmetric fashion. Insertions occuring between the same residues in both the child and the parent representative sequence are considered to have been independently acquired. Since these insertions are non homologous they cannot be aligned and are therefore concatenated (i.e.

blocks of gaps are inserted in the child to match the parent insertion, and in the parent to match the child insertion). This merging is linear in time and memory and proportional to the unaligned length of the two sequences. The RAM memory footprint is further lowered by recording the length of gap indels and inserting them only when writing the final MSA onto disk.

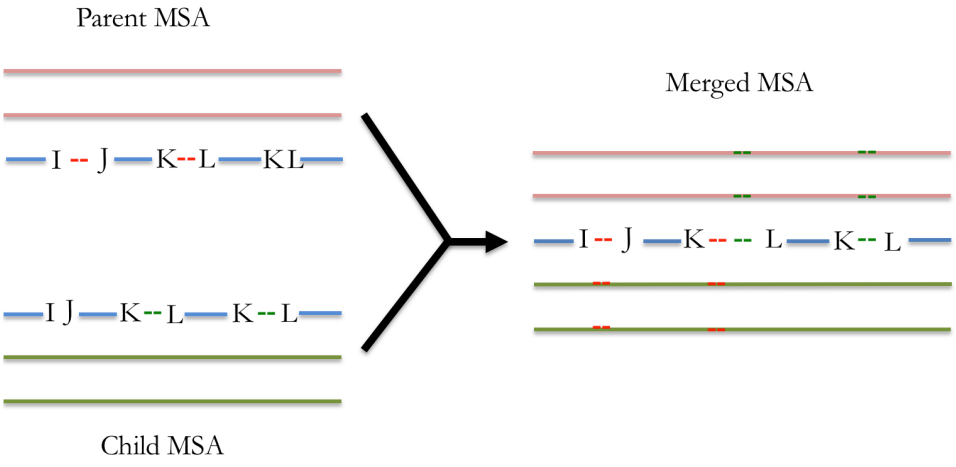


Figure 1a. The merging of a child and parent MSA and made possible through representative sequences shown here in blue without the need for profile-profile alignments.

A key step of this recursion is the selection of N representative sequences within a dataset of M sequences. K-means or any related algorithms could provide a simple and efficient way to produce these groups, yet for the sake of benchmarking and comparison with existing methods, we chose to generate the representative sets from third party binary guide trees generated by large scale aligners (Figure 1b). Under this scheme, each node is assigned the label of the longest of the two sequence labeling its left and right

children. Starting from the leaves, every parent node is therefore labeled with the name of its longest child sequence and the algorithm proceeds accordingly all the way up to the root, labeled with the longest sequence. Given a fully labeled tree, the sequences of the first parent MSA are collected by expanding the root, its children, and the next generation children iteratively until N sequences have been collected. Since these N nodes correspond to as many non-overlapping children subtrees, each node effectively provides both a representative (the node label) and a sub-dataset (all the leaves connected to this node). Within the first parent MSA, each sequence is either a leaf or an internal node label. Internal nodes are recursively processed in a similar way until all leaves have been incorporated in an MSA. Once all the MSAs have been collected they are merged into the final MSA.

As defined above, the regressive algorithm does not depend on a specific alignment procedure. This enabled us to use third party aligners for both guide tree generation and the computation of parent and children MSAs. By keeping all things equal aside from the agglomerative procedure this approach therefore provides a direct estimate of the progressive and regressive algorithm relative accuracy. We used this approach to systematically compare ClustalO and Mafft using the ClustalO embed k-means and Maftt parttrees as guide trees. These two aligners were selected because they are strictly progressive and allow input and and output of binary guide trees.

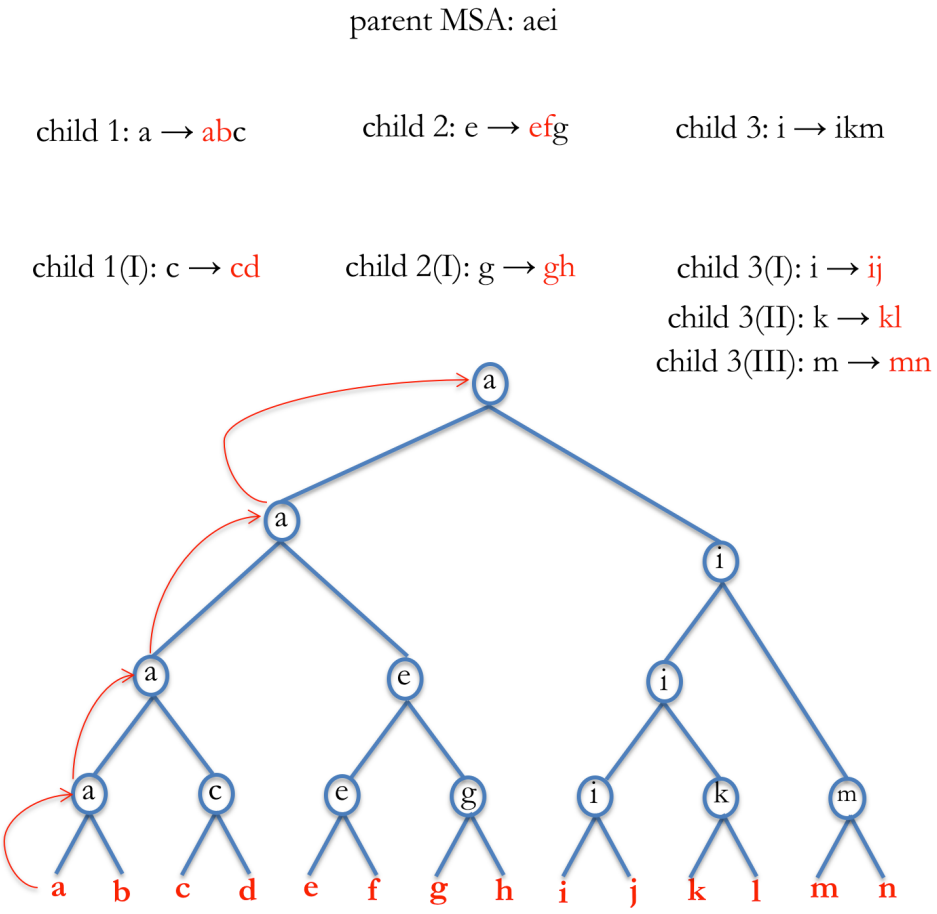


Figure 1b. The guide tree is initially labelled with the longest sequence of each child node. Starting with the root node, the sequences of the parent MSA are collected by expanding the root, the children, and the next generation children iteratively until N sequences have been collected. In the example above, with N=3, the parent MSA would consist of sequences a, e and i. From this MSA, the internal nodes are recursively expanded in a similar way until all sequences at the leaves have been added to a MSA. Once all the MSA have been generated (7 MSAs in the example above), they are merged through the common representative sequence.

Tree Method	*MSA method*	Total Column Score (TC)				
		Non Regressive		**Regressive**		**Reference**
		Score (%)	**Rel. Score**	**Score (%)**	**Rel. Score**	**Score (%)**
parttree	Mafft-fftns1	29.64	0.62	35.97	0.75	47.84
clustalo	Mafft-fftns1	41.33	0.79	40.95	0.79	52.03
parttree	clustalo	26.94	0.53	42.21	0.84	50.54
clustalo	clustalo	39.03	0.73	41.91	0.78	53.71
Average		**34.24**	**0.67**	**40.26**	**0.79**	**51.03**
default/clustalo	upp	43.61	0.87	44.15	0.89	49.85
default/clustalo	Mafft-sparsecore	44.98	0.84	51.07	0.95	53.51
parttree	Mafft-ginsi	-		47.54	0.96	49.46
clustalo	Mafft-ginsi	-		50.21	0.95	53.07
Global Average		**37.59**	**0.73**	**42.71**	**0.86**	**51.25**

Table 1: Total column score (TC) on the largest 20 datasets from the HOMFAM benchmarking dataset.

In three out of four combinations of tree and aligner, the regressive implementation outperforms the progressive and when considering the most discriminative measure (total column score, TC). On the 20 largest datasets, the regressive algorithm delivers MSAs that are 6.5 points more accurate than when estimated in a progressive manner (40.26 and 34.24 respectively). Out of all these combinations, the most accurate on large datasets is the regressive implementation of ClustalO using parttree that outperforms its progressive equivalent by 15.27 points (42.21 and 26.94 respectively). We used PCA to dissect the contribution of each component in these analysis with a clear indication of improved accuracy being driven by the regressive algorithm. We did the same analysis on UPP and MAFFT-sparsecore, the two most recent large scale aligners. While the standalone version of these aligners is significantly more accurate than ClustalO and MAFFT-sparsecore, we were able to show that the regressive deployment of MAFFT-sparsecore using a ClustalO guide tree outperforms all alternative protocols evaluated here (51.07 vs 44.98 for Mafft-sparsecore, the best aligner in non-regressive mode)

Albeit clearly superior to its progressive counterpart, the regressive assembly nonetheless fails at preventing the accuracy drop associated with sequences embedding. We therefore took advantage of the regressive algorithm modular nature to go one step further and combined methods that were not initially meant to be so. For instance, ginsi, the consistency based flavor of

Mafft, is among the most accurate small scale aligner on the reference sequences but the cost of the consistency transformation, cubic in time with the number of sequences, prevents it from aligning over a thousand sequences. This limitation is easily overcome by deploying ginsi in a regressive way. We did so using both the ClustalO and the PartTree guidetrees and found these combinations to result in some of the most accurate models reported across all the analysis carried out here (Table 1). On the 20 largest datasets, the best regressive ginsi mode is 8.88 points better that the best progressive aligner (Mafft-fftns1 with a ClustalO tree) and 5.23 points better that the best seeded aligner (Mafft-sparsecore). Even more importantly our analysis show that the regressive deployment of ginsi is one of the method less affected by the scaling up when considering the drop in accuracy with respect to the direct MSA of the reference sequences.
20 points more accurate than the progressive aligners, and 11.9 points over the single best progressive aligner (Regressive-ginsi vs progressive MAFFT with ClustalO guide trees for both). Even more important that the absolute accuracy, we show that the most accurate flavor of the regressive ginsi is almost not affected by the scaling up (68.82 on the seed MSAs vs 68.32 on the embedded sequences).

The ginsi improvement comes at cost with CPU requirement almost two orders of magnitude above fftns1, the fastest method benchmarked here. This overhead is, however, manageable thanks to the high order parallelisation allowed by the precomputation of the parent and children

alignments. The lack of any dependence between these models makes it possible to estimate them all at the same time and their merging is linear in time with both the length and the number of sequences. The scaling up also appears to be slightly more favorable for the regressive implementation, especially when dealing with the most CPU demanding datasets. The comparison is even more favorable to the regressive approach when considering identical aligners for which the regressive performances often outperform the progressive agglomeration.

Altogether these results suggest that the regressive approach described here provides a practical solution to the critical problem of MSA scalability - a problem fueled by the accelerating pace of high throughput whole genome sequencing. Not only does the regressive approach provide a mature solution, but it also defines a very new exciting development framework by providing a clean break between the development of highly accurate small scale aligners - like ginisi - and the design of novel scale clustering algorithms, like parttree and ClustalO. Until now, the aligner and the clustering algorithm had been tightly connected with each component fine-tuned to compensate the weakness of the other and extra iterations meant to fix everything. The regressive framework alleviates these constraints and allows the independent combination of all available methods even when these involve iterative refinements. But the the regressive mode is also more general that the progressive as it is less strictly bound to a binary pre-clustering and could be deployed using k-trees and even b-trees whose

node order may vary. Exploring the possibilities afforded by these many variation will be the focus of further studies in the longterm quest for scalable multiple sequence alignment comparisons.

REFERENCES

Boyce, Kieran, Fabian Sievers, and Desmond G. Higgins. 2014. "Simple Chained Guide Trees Give High-Quality Protein Multiple Sequence Alignments." *Proceedings of the National Academy of Sciences of the United States of America* 111 (29): 10556–61.

Katoh, K. 2002. "MAFFT: A Novel Method for Rapid Multiple Sequence Alignment Based on Fast Fourier Transform." *Nucleic Acids Research* 30 (14): 3059–66.

Nguyen, Nam-Phuong D., Siavash Mirarab, Keerthana Kumar, and Tandy Warnow. 2015. "Ultra-Large Alignments Using Phylogeny-Aware Profiles." *Genome Biology* 16 (June): 124.

Sievers, Fabian, and Desmond G. Higgins. 2013. "Clustal Omega, Accurate Alignment of Very Large Numbers of Sequences." In *Methods in Molecular Biology*, 105–16.

Tan, Ge, Manuel Gil, Ari P. Löytynoja, Nick Goldman, and Christophe Dessimoz. 2015. "Simple Chained Guide Trees Give Poorer Multiple Sequence Alignments than Inferred Trees in Simulation and Phylogenetic Benchmarks." *Proceedings of the National Academy of Sciences of the United States of America* 112 (2): E99–100.

Van Noorden, Richard, Brendan Maher, and Regina Nuzzo. 2014. "The Top

100 Papers." *Nature* 514 (7524): 550–53.

Yamada, Kazunori D., Kentaro Tomii, and Kazutaka Katoh. 2016. "Application of the MAFFT Sequence Alignment Program to Large Data-Reexamination of the Usefulness of Chained Guide Trees." *Bioinformatics* 32 (21): 3246–51.

Chapter 6

Nextflow enables reproducible computational workflows

Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C.

Nat Biotechnol. 2017 Apr 11;35(4):316-319. DOI: 10.1038/nbt.3820

7. Chapter 7: Discussion

7.1 PSI/TM-Coffee webserver

When performing sequence searches, as the pairwise identity of proteins drops below approximately 30%, the number of false negatives explodes and many true homologous sequences are missed (Rost 1999). In the Chapter 1 we discussed PSI-BLAST which uses profiles created from queries to perform BLAST homology searches. We can apply the same principle to multiple sequence alignment. With PSI-Coffee, for every sequence in to our dataset, we first perform a BLAST search and collect the resulting sequences (Chang et al. 2012). Each collection of sequences - one set per sequence in the original dataset - is then transformed into a profile . Profiles prove to be a valuable tool in homology detection and are especially useful when the identity of homologues is in the twilight zone. With homology extension, the evolutionary constraints at each position in the query sequences are examined and quantified. By aligning these profiles and not the sequences themselves, the quality of pairwise alignments is improved leading to a reduction in highly detrimental errors early in the alignment process.

In the homology extension step, PSI-Coffee uses BLAST to search for homologous sequences against a full database of sequences. However in some situations, it may be more efficient to search against a reduced database. This is particularly applicable to transmembrane proteins (TMPs) which make up 20 and 30% of prokaryotic and eukaryotic proteins. TMPs,

whose structure traverses the entire membrane, are central players in many important biological processes. They act as gateways for the transport of specific molecules. This has lead to significant research efforts into understanding their function and exploitation as potential as drug targets. It has proven exceptionally hard to experimentally determine the 3D structure of TMPs due to difficulties in purification and crystallisation. This places even more emphasis on homology based prediction methods.

Chapter 2 describes the web server for PSI/TM-Coffee which uses a reduced UniRef database that is filtered to contain only TMPs. This reduced database is shown to obtain similar results at a significantly reduced computational cost over full protein databases. In evaluating the performance of the method with BAliBASE2-ref7 α-helical TMPs, PSI/TM-Coffee displayed a significant improvement in comparison to the most accurate methods (MSAProbs, Kalign, PROMALS, MAFFT, ProbCons and PRALINE).

The PSI/TM-Coffee web server itself has become an integral part of the T-Coffee family of online resources. It has contributed to the widespread adoption of T-Coffee web server by the many communities (Di Tommaso et al. 2011). The overall usage of the T-Coffee service shows approximately 70,000 unique users in the last 12 months with the original web server publication having approximately 500 citations since 2011. This alone shows

the importance of these services and their significance to users whom rely on them to perform accurate and efficient multiple sequence alignments.

7.3 Phylogenetic supports incorporating alignment uncertainty

Phylogenies are one of the most common inferences made from MSA methods. The large number of high impact works published in the field over these last years highlights the need of the biological community for reliable methods. Indeed phylogeny reconstruction is often an essential step for the generation of evolutionary hypotheses.

Given a phylogenetic tree, branch support analysis is, for the most part, currently carried out using Felsenstein's bootstrap method. This procedure is common in most phylogenetic studies and has received nearly 10,000 citations over the last 30 years. By incorporating alignment uncertainty, we show that the original bootstrap measure does not capture all the confounding factors associated with tree building. In chapters 3 and 4 I describe two new methods that attempt to capture these effects and can be used to estimate branch stability in phylogenetic trees.

In chapter 3 *Using alignment uncertainty to improve phylogenetic bootstrap reliability* I describe an approach that builds on the work of Wong et al. published in Science in 2008. In this work it was established how uncertainty from multiple alignment procedures effects reconstructing phylogenies and

inferring evolutionary rates. They were able to show that in many cases different aligners produce different phylogenies, with no simple objective criterion sufficient to distinguish among these alternatives. They did however, stop short of proposing a solution. Indeed, it is relatively easy to tell apart two alternative trees based on the same alignment, but it is much less straightforward to determine the relative merits of two or more alternative alignments and their associated phylogenies. When building phylogenies, one is left with no option but to use the methods reported to be on average the most accurate on one benchmark or another.

With this first method we propose a simple but effective solution to incorporate the uncertainty and instability generated by the various alternative alignment methods. This way, one does not need anymore to arbitrarily choose an alignment method. In fact, we show how the combination of these uncertainties adds up into significantly more informative bootstrap values and therefore ends up increasing the level of certainty. This approach does not appear to yield better trees but it increases dramatically the capacity to discriminate between correct and incorrect trees.

In chapter 4 *Generalized bootstrap supports for phylogenetic analyses of protein sequences incorporating alignment uncertainty* I describe a different approach to the same problem. This work first establishes that all available large-scale aligners, including the most recent ones, can be induced to produce very unstable alignment models and phylogenetic trees by simply changing

sequences input order. Considering the lack of any objective criteria to define an optimal sequence input-order - that may not even exist - this finding is problematic for most analysis. In the benchmarks performed, instability dominates large datasets with less than 50% of the branches being reproducible when dealing dataset consisting of 10,000 sequences or more. Branches are affected across the entire trees, including deep and shallow nodes. Even though we demonstrate that no solution exists to solve this problem, we nonetheless show how alignment induced instability can be used to estimate a new branch reliability index. This index was initially a combination of the regular bootstrap estimate procedure that quantifies column sampling effects with the input order effect so as to provide a combined estimation of the joint effect of column sampling and taxa shuffling onto each branch in the final tree. However it became apparent that it could be generalised to take alternative alignments from any source and was thus named Unistrap.

These works together describes a simple and effective methods to quantify this effect of alignment instability onto phylogenetic tree reconstruction. They provide the biological community with novel conceptual tools that allow proper quantification of all the confounding factors affecting tree reconstruction, including MSA induced noise and evolutionary sampling.

7.4 The Regressive Multiple Sequence Alignment

Multiple sequence alignments are essential for a large number of tasks in biology including phylogenetic inference, structural modeling and functional predictions. The increase in the size of the datasets used in these applications necessitates methods that likewise scale. In chapter 5, *Regressive computation of large scale multiple sequence alignments*, I describe a new agglomerative multiple sequence alignment algorithm whose scaling up capacities outperform all available methods in terms of accuracy.

The computation of accurate multiple sequence alignments is an NP-complete problem. There is no exact solution guaranteed and for this reason all available methods are based on approximate heuristics. Reliance on heuristics requires these methods to be revisited and readapted each time the nature of the problem changes, even slightly. For instance, over the last years, the growing appetite for increasingly large datasets has revealed an unforeseen limitation of the current alignment framework - known as progressive alignment. Against all expectations, alignment accuracy decreases when increasing the number of sequences above a thousand homologues. This result was a genuine surprise because it had long been observed that all things being equal, a given group of sequences would see its relative alignment accuracy increase when embedded within a larger dataset. This limitation is a major issue because it brings the current paradigm of MSA scaling up to a dead-end. It casts serious doubts on our

capacity to effectively integrate the biological information contributed by the new genome projects.

I present a very simple and extremely effective way to scale up MSA modelling methods termed regressive by reference to the progressive algorithm. When doing a progressive (or a regressive) alignment, sequences are clustered using a guide tree that defines the order in which they will be aligned. The progressive alignments then starts by aligning the most similar sequences - sister leafs - and proceeds all the way until the root. The regressive approach uses the same tree, but rather than going from leaf to root, we first use the tree to collect the most diverse sequences and then start aligning them, the same algorithm is then applied recursively while proceeding towards the root. The first alignment that contains the most diverse sequences is treated as a scaffold onto which all subsequent alignments are grafted.

For validation purpose, the implemented the algorithm allows for use of common large scale aligners - Mafft, ClustalO and UPP to be deployed in both a regressive and a progressive way. This approach allowed us to dissect precisely the contribution of each algorithmic component and conclude on the superiority of the regressive approach over the progressive one. All things being equal, on the 20 largest reference datasets in HOMFAM, (10,000 to 93,000 sequences), the regressive approach outperforms the progressive approach by over 6.5 percentage points on average. More

importantly, the improved scalability of the regressive framework also allowed us to deploy small scale highly accurate methods like mafft-ginsi on very large datasets for which they were not originally intended. The resulting alignments are the most accurate ever reported on these datasets. This result is of direct practical use to the community since this validation, comes along with a mature production software implemented in T-Coffee and available on GitHub.

The regressive algorithm is, however, much more than a new software. Thanks to the clear separation it provides between the guide tree and the aligners, the regressive algorithm redefines the field of research in multiple sequence alignment computation. It allows a strict dichotomy to be implemented between the development of highly accurate small scale aligners on one side and the development of ever faster and more accurate clustering algorithms on the other side. By explicitly breaking the connection between alignment and clustering, it is hoped the two independent communities can contribute their specific capacities and develop novel methods whose availability is of strategic importance for the future of biology.

7.5 The Nextflow Workflow Framework

At a time when the precision medicine initiative is about to introduce the systematic use of -omics data in our everyday life, the notion of reproducible genomic analysis appears more critical than ever. It is often assumed that

reproducibility issues merely result from wet lab experimental fluctuation. In chapter 6 I show that this assumption is incorrect and that standard in-silico analysis - such as RNA-Seq quantification and phylogenetic reconstruction - can be substantially unstable across the most common computational platforms, even when using state-of-the art genomic analysis tools. Nextflow is a method for computational workflow management that provides a simple and effective solution to this problem. It is shown how Nextflow makes it possible to deploy existing pipelines in an efficient and stable fashion and provides a long awaited answer to the issue of guaranteeing computational reproducibility when running -omics data analysis.

The principles developed in Nextflow are appealing to anyone developing high throughput data analysis pipelines with limited software development resources. While most existing similar framework, such as Galaxy, require full pipeline re-implementation, Nextflow is a light weight solution that makes it possible to rapidly adapt any third party tool with limited re-coding requirements. Once adapted, tools can be deployed agnostically across the most common IT infrastructures, like clouds, supercomputers, local clusters and workstations. Nextflow is a freeware open-source software and has been designed to fuel collaboration and help efficiently compare alternative numerical analysis procedures in an open way.

It is already a mature solution with a growing community of users with extensive documentation and support. Nextflow has been adopted into daily

use by scientists in companies and research institute alike such as the Broad Institute, the Joint Genome Institute, Cornell University, The Sanger Institute, SciLifeLab, Karolinska Institute and the International Agency for Research in Cancer among others. This shows how important and timely the contribution is and it is hoped Nextflow will have a long lasting impact on the establishment of novel quality standards for reproducible Big Data analysis, in biology and beyond.

Bibliography

Afgan, Enis, Dannon Baker, Marius van den Beek, Daniel Blankenberg, Dave Bouvier, Martin Čech, John Chilton, et al. 2016. "The Galaxy Platform for Accessible, Reproducible and Collaborative Biomedical Analyses: 2016 Update." *Nucleic Acids Research* 44 (W1): W3–10.

Altschul, S. F., and W. Gish. 1996. "Local Alignment Statistics." *Methods in Enzymology* 266: 460–80.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. "Basic Local Alignment Search Tool." *Journal of Molecular Biology* 215 (3): 403–10.

Altschul, S. F., T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. 1997. "Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs." *Nucleic Acids Research* 25 (17): 3389–3402.

Andrade, M. A. 2003. *Bioinformatics and Genomes: Current Perspectives*. Taylor & Francis.

Anfinsen, Christian B. 1959. *The Molecular Basis of Evolution*.

Armougom, Fabrice, Sébastien Moretti, Olivier Poirot, Stéphane Audic, Pierre Dumas, Basile Schaeli, Vladimir Keduas, and Cedric Notredame. 2006. "Expresso: Automatic Incorporation of Structural Information in Multiple Sequence Alignments Using 3D-Coffee." *Nucleic Acids Research* 34 (Web Server issue): W604–8.

Baker, W. 2000. "The EMBL Nucleotide Sequence Database." *Nucleic Acids Research* 28 (1): 19–23.

Bernstein, F. C., T. F. Koetzle, G. J. Williams, E. F. Meyer Jr, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. 1977. "The

Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures." *Journal of Molecular Biology* 112 (3): 535–42.

Blackshields, Gordon, Fabian Sievers, Weifeng Shi, Andreas Wilm, and Desmond G. Higgins. 2010. "Sequence Embedding for Fast Construction of Guide Trees for Multiple Sequence Alignment." *Algorithms for Molecular Biology: AMB* 5 (May): 21.

Blankenberg, Daniel, Gregory Von Kuster, Emil Bouvier, Dannon Baker, Enis Afgan, Nicholas Stoler, Galaxy Team, James Taylor, and Anton Nekrutenko. 2014. "Dissemination of Scientific Software with Galaxy ToolShed." *Genome Biology* 15 (2): 403.

Boettiger, Carl. 2015. "An Introduction to Docker for Reproducible Research." *ACM SIGOPS Operating Systems Review* 49 (1): 71–79.

Capella-Gutiérrez, Salvador, José M. Silla-Martínez, and Toni Gabaldón. 2009. "trimAl: A Tool for Automated Alignment Trimming in Large-Scale Phylogenetic Analyses." *Bioinformatics* 25 (15): 1972–73.

Castresana, J. 2000. "Selection of Conserved Blocks from Multiple Alignments for Their Use in Phylogenetic Analysis." *Molecular Biology and Evolution* 17 (4): 540–52.

Chang, Jia-Ming, Paolo Di Tommaso, and Cedric Notredame. 2014. "TCS: A New Multiple Sequence Alignment Reliability Measure to Estimate Alignment Accuracy and Improve Phylogenetic Tree Reconstruction." *Molecular Biology and Evolution* 31 (6): 1625–37.

Chang, Jia-Ming, Paolo Di Tommaso, Jean-François Taly, and Cedric Notredame. 2012. "Accurate Multiple Sequence Alignment of Transmembrane Proteins with PSI-Coffee." *BMC Bioinformatics* 13 Suppl 4 (March): S1.

Chargaff, E., and B. Magasanik. 1949. "The Nucleotide Composition of

Ribonucleic Acids." *Journal of the American Chemical Society* 71 (4): 1513.

Consden, R., A. H. Gordon, and A. J. Martin. 1947. "Gramicidin S: The Sequence of the Amino-Acid Residues." *Biochemical Journal* 41 (4): 596–602.

Cummings, Michael P. 2004. "MrBayes." In *Dictionary of Bioinformatics and Computational Biology*.

Curtius, Theodor. 1883. "Ueber Die Einwirkung von Salpetriger Säure Auf Salzsauren Glycocolläther." *Berichte Der Deutschen Chemischen Gesellschaft* 16 (2): 2230–31.

Di Tommaso, Paolo, Sebastien Moretti, Ioannis Xenarios, Miquel Orobitg, Alberto Montanyola, Jia-Ming Chang, Jean-François Taly, and Cedric Notredame. 2011. "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 (Web Server issue): W13–17.

Di Tommaso, Paolo, Emilio Palumbo, Maria Chatzou, Pablo Prieto, Michael L. Heuer, and Cedric Notredame. 2015. "The Impact of Docker Containers on the Performance of Genomic Pipelines." *PeerJ* 3 (September): e1273.

Do, Chuong B., Mahathi S. P. Mahabhashyam, Michael Brudno, and Serafim Batzoglou. 2005. "ProbCons: Probabilistic Consistency-Based Multiple Sequence Alignment." *Genome Research* 15 (2): 330–40.

Drummond, Alexei J., and Andrew Rambaut. 2007. "BEAST: Bayesian Evolutionary Analysis by Sampling Trees." *BMC Evolutionary Biology* 7 (1): 214.

Dumas, Jean-Pierre, and Jacques Ninio. 1982. "Efficient Algorithms for Folding and Comparing Nucleic Acid Sequences." *Nucleic Acids Research*

10 (1): 197–206.

Durbin, Richard, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Eddy, S. R. 1998. "Profile Hidden Markov Models." *Bioinformatics* 14 (9): 755–63.

Edgar, Robert C. 2004. "MUSCLE: Multiple Sequence Alignment with High Accuracy and High Throughput." *Nucleic Acids Research* 32 (5): 1792–97.

Efron, B., E. Halloran, and S. Holmes. 1996. "Bootstrap Confidence Levels for Phylogenetic Trees." *Proceedings of the National Academy of Sciences of the United States of America* 93 (23): 13429–34.

Farris, J. S. 1970. "Methods for Computing Wagner Trees." *Systematic Biology* 19 (1): 83–92.

Felsenstein, Joseph. 1981. "Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach." *Journal of Molecular Evolution* 17 (6): 368–76.

Felsenstein, Joseph. 1985. "Confidence Limits On Phylogenies: An Approach Using The Bootstrap." *Evolution; International Journal of Organic Evolution* 39 (4): 783–91.

Fitch, Walter M. 1971. "Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology." *Systematic Zoology* 20 (4): 406.

Fitch, W. M., and K. T. Yasunobu. 1975. "Phylogenies from Amino Acid Sequences Aligned with Gaps: The Problem of Gap Weighting." *Journal of Molecular Evolution* 5 (1): 1–24.

Gatesy, J., R. DeSalle, and W. Wheeler. 1993. "Alignment-Ambiguous

Nucleotide Sites and the Exclusion of Systematic Data." *Molecular Phylogenetics and Evolution* 2 (2): 152–57.

Gotoh, O. 1982. "An Improved Algorithm for Matching Biological Sequences." *Journal of Molecular Biology* 162 (3): 705–8.

Gotoh, O.. 1993. "Optimal Alignment between Groups of Sequences and Its Application to Multiple Sequence Alignment." *Computer Applications in the Biosciences: CABIOS* 9 (3): 361–70.

Grüning, Björn, John Chilton, Johannes Köster, Ryan Dale, Nicola Soranzo, Marius van den Beek, Jeremy Goecks, Rolf Backofen, Anton Nekrutenko, and James Taylor. 2018. "Practical Computational Reproducibility in the Life Sciences." *Cell Systems* 6 (6): 631–35.

Guindon, Stéphane, Jean-François Dufayard, Vincent Lefort, Maria Anisimova, Wim Hordijk, and Olivier Gascuel. 2010. "New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0." *Systematic Biology* 59 (3): 307–21.

Haeckel, Ernst. 1866. *Generelle Morphologie der Organismen: Bd. Allgemeine Entwickelungsgeschichte der Organismen*.

Henikoff, S., and J. G. Henikoff. 1992. "Amino Acid Substitution Matrices from Protein Blocks." *Proceedings of the National Academy of Sciences of the United States of America* 89 (22): 10915–19.

Higgins, D. G., and P. M. Sharp. 1988. "CLUSTAL: A Package for Performing Multiple Sequence Alignment on a Microcomputer." *Gene* 73 (1): 237–44.

Hirosawa, M., Y. Totoki, M. Hoshida, and M. Ishikawa. 1995. "Comprehensive Study on Iterative Algorithms of Multiple Sequence Alignment." *Computer Applications in the Biosciences: CABIOS* 11 (1): 13–18.

Hofmeister, F. 1902. "Über Bau Und Gruppierung Der Eiweisskörper." *Ergebnisse Der Physiologie, Biologischen Chemie Und Experimentellen Pharmakologie* 1 (1): 759–802.

Hogeweg, P., and B. Hesper. 1984. "The Alignment of Sets of Sequences and the Construction of Phyletic Trees: An Integrated Method." *Journal of Molecular Evolution* 20 (2): 175–86.

Ingram, Vernon M. 1961. "Gene Evolution and the Hæmoglobins." *Nature* 189 (4766): 704–8.

Jacob, F. 1977. "Evolution and Tinkering." *Science* 196 (4295): 1161–66.

Katoh, Kazutaka, Kazuharu Misawa, Kei-Ichi Kuma, and Takashi Miyata. 2002. "MAFFT: A Novel Method for Rapid Multiple Sequence Alignment Based on Fast Fourier Transform." *Nucleic Acids Research* 30 (14): 3059–66.

Katoh, Kazutaka, and Hiroyuki Toh. 2007. "PartTree: An Algorithm to Build an Approximate Tree from a Large Number of Unaligned Sequences." *Bioinformatics* 23 (3): 372–74.

Köster, Johannes, and Sven Rahmann. 2012. "Snakemake--a Scalable Bioinformatics Workflow Engine." *Bioinformatics* 28 (19): 2520–22.

Kreitman, M. 1983. "Nucleotide Polymorphism at the Alcohol Dehydrogenase Locus of Drosophila Melanogaster." *Nature* 304 (5925): 412–17.

Kumar, Sudhir, and Alan Filipski. 2004. "Neighbor-Joining Method." In *Dictionary of Bioinformatics and Computational Biology*.

Kurtzer, Gregory M., Vanessa Sochat, and Michael W. Bauer. 2017. "Singularity: Scientific Containers for Mobility of Compute." *PloS One* 12 (5): e0177459.

Lake, J. A. 1991. "The Order of Sequence Alignment Can Bias the Selection of Tree Topology." *Molecular Biology and Evolution* 8 (3): 378–85.

Landan, Giddy, and Dan Graur. 2007. "Heads or Tails: A Simple Reliability Check for Multiple Sequence Alignments." *Molecular Biology and Evolution* 24 (6): 1380–83.

Lipman, D. J., and W. R. Pearson. 1985. "Rapid and Sensitive Protein Similarity Searches." *Science* 227 (4693): 1435–41.

Löytynoja, Ari. 2014. "Phylogeny-Aware Alignment with PRANK." *Methods in Molecular Biology* 1079: 155–70.

Malhotra, Raunaq, Isheeta Seth, Erik Lehnert, Jing Zhao, Gaurav Kaushik, Elizabeth H. Williams, Anurag Sethi, and Brandi N. Davis-Dusenbery. 2017. "Using the Seven Bridges Cancer Genomics Cloud to Access and Analyze Petabytes of Cancer Data." *Current Protocols in Bioinformatics / Editoral Board, Andreas D. Baxevanis ... [et Al.]* 60 (December): 11.16.1–11.16.32.

Margush, T., and F. R. McMorris. 1981. "Consensusn-Trees." *Bulletin of Mathematical Biology* 43 (2): 239–44.

McLachlan, A. D. 1971. "Tests for Comparing Related Amino-Acid Sequences. Cytochrome c and Cytochrome c551." *Journal of Molecular Biology* 61 (2): 409–24.

Morrison, D. A., and J. T. Ellis. 1997. "Effects of Nucleotide Sequence Alignment on Phylogeny Estimation: A Case Study of 18S rDNAs of Apicomplexa." *Molecular Biology and Evolution* 14 (4): 428–41.

Mount, David W. 2009. "Using Hidden Markov Models to Align Multiple Sequences." *Cold Spring Harbor Protocols* 2009 (7): db.top41.

Mueller, L. D., and F. J. Ayala. 1982. "Estimation and Interpretation of

Genetic Distance in Empirical Studies." *Genetical Research* 40 (2): 127–37.

Myers, Eugene W., and Webb Miller. 1988. "Optimal Alignments in Linear Space." *Bioinformatics* 4 (1): 11–17.

Nawrocki, Eric P., Sarah W. Burge, Alex Bateman, Jennifer Daub, Ruth Y. Eberhardt, Sean R. Eddy, Evan W. Floden, et al. 2015. "Rfam 12.0: Updates to the RNA Families Database." *Nucleic Acids Research* 43 (Database issue): D130–37.

Nawrocki, Eric P., and Sean R. Eddy. 2013. "Infernal 1.1: 100-Fold Faster RNA Homology Searches." *Bioinformatics* 29 (22): 2933–35.

Needleman, S. B., and C. D. Wunsch. 1970. "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins." *Journal of Molecular Biology* 48 (3): 443–53.

Neel, J. V. 1952. "Perspectives in the Genetics of Sickle Cell Disease." *Blood* 7 (4): 467–71.

Nguyen, Nam-Phuong D., Siavash Mirarab, Keerthana Kumar, and Tandy Warnow. 2015. "Ultra-Large Alignments Using Phylogeny-Aware Profiles." *Genome Biology* 16 (June): 124.

Nirenberg, M. W., and J. H. Matthaei. 1961. "The Dependence of Cell-Free Protein Synthesis in E. Coli upon Naturally Occurring or Synthetic Polyribonucleotides." *Proceedings of the National Academy of Sciences of the United States of America* 47 (October): 1588–1602.

Notredame, C., D. G. Higgins, and J. Heringa. 2000. "T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment." *Journal of Molecular Biology* 302 (1): 205–17.

Pauling, L., and H. A. Itano. 1949. "Sickle Cell Anemia a Molecular Disease." *Science* 110 (2865): 543–48.

Pearson, W. R., and D. J. Lipman. 1988. "Improved Tools for Biological Sequence Comparison." *Proceedings of the National Academy of Sciences of the United States of America* 85 (8): 2444–48.

Penn, O., E. Privman, H. Ashkenazy, G. Landan, D. Graur, and T. Pupko. 2010. "GUIDANCE: A Web Server for Assessing Alignment Confidence Scores." *Nucleic Acids Research* 38 (Web Server): W23–28.

Pulverer, Bernd. 2015. "Data Accessibility and Reproducibility: Moving to Transparent Publishing in the Biosciences." *Information Services & Use* 35 (3): 185–88.

Sanger, F., G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, J. C. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith. 1977. "Nucleotide Sequence of Bacteriophage ϕX174 DNA." *Nature* 265 (5596): 687–95.

Sanger, F., S. Nicklen, and A. R. Coulson. 1977. "DNA Sequencing with Chain-Terminating Inhibitors." *Proceedings of the National Academy of Sciences of the United States of America* 74 (12): 5463–67.

Sanger, F., and H. Tuppy. 1951. "The Amino-Acid Sequence in the Phenylalanyl Chain of Insulin. I. The Identification of Lower Peptides from Partial Hydrolysates." *Biochemical Journal* 49 (4): 463–81.

Sankoff, D. 1972. "Matching Sequences under Deletion-Insertion Constraints." *Proceedings of the National Academy of Sciences of the United States of America* 69 (1): 4–6.

Sarich, V. M., and A. C. Wilson. 1967. "Immunological Time Scale for Hominid Evolution." *Science* 158 (3805): 1200–1203.

Sela, Itamar, Haim Ashkenazy, Kazutaka Katoh, and Tal Pupko. 2015. "GUIDANCE2: Accurate Detection of Unreliable Alignment Regions Accounting for the Uncertainty of Multiple Parameters." *Nucleic Acids*

Research 43 (W1): W7–14.

Sellers, Peter H. 1974. "On the Theory and Computation of Evolutionary Distances." *SIAM Journal on Applied Mathematics* 26 (4): 787–93.

Sievers, Fabian, and Desmond G. Higgins. 2013. "Clustal Omega, Accurate Alignment of Very Large Numbers of Sequences." In *Methods in Molecular Biology*, 105–16.

Smith, T. F., and M. S. Waterman. 1981. "Identification of Common Molecular Subsequences." *Journal of Molecular Biology* 147 (1): 195–97.

Sokal, Robert Reuven, Charles Duncan Michener, and University of Kansas. 1958. *A Statistical Method for Evaluating Systematic Relationships*.

Sonnhammer, E. L., S. R. Eddy, E. Birney, A. Bateman, and R. Durbin. 1998. "Pfam: Multiple Sequence Alignments and HMM-Profiles of Protein Domains." *Nucleic Acids Research* 26 (1): 320–22.

Stamatakis, Alexandros. 2014. "RAxML Version 8: A Tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies." *Bioinformatics* 30 (9): 1312–13.

States, D., W. Gish, and S. Altschul. 1991. "Improved Sensitivity of Nucleic Acid Database Searches Using Application-Specific Scoring Matrices." *Methods* 3 (1): 66–70.

Strasser, Bruno J. 2010. "Collecting, Comparing, and Computing Sequences: The Making of Margaret O. Dayhoff's Atlas of Protein Sequence and Structure, 1954-1965." *Journal of the History of Biology* 43 (4): 623–60.

Stretton, Antony O. W. 2002. "The First Sequence. Fred Sanger and Insulin." *Genetics* 162 (2): 527–32.

Thompson, J. D., D. G. Higgins, and T. J. Gibson. 1994. "CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment

through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice." *Nucleic Acids Research* 22 (22): 4673–80.

Thorne, J. L., H. Kishino, and J. Felsenstein. 1991. "An Evolutionary Model for Maximum Likelihood Alignment of DNA Sequences." *Journal of Molecular Evolution* 33 (2): 114–24.

Van Noorden, Richard, Brendan Maher, and Regina Nuzzo. 2014. "The Top 100 Papers." *Nature* 514 (7524): 550–53.

Wallace, Iain M., Orla O'Sullivan, Desmond G. Higgins, and Cedric Notredame. 2006. "M-Coffee: Combining Multiple Sequence Alignment Methods with T-Coffee." *Nucleic Acids Research* 34 (6): 1692–99.

Watson, J. D., and F. H. C. Crick. 1953. "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid." *Nature* 171 (4356): 737–38.

Wheeler, W. C., J. Gatesy, and R. DeSalle. 1995. "Elision: A Method for Accommodating Multiple Molecular Sequence Alignments with Alignment-Ambiguous Sites." *Molecular Phylogenetics and Evolution* 4 (1): 1–9.

Wilbur, W. J., and D. J. Lipman. 1983. "Rapid Similarity Searches of Nucleic Acid and Protein Data Banks." *Proceedings of the National Academy of Sciences of the United States of America* 80 (3): 726–30.

Wilm, Andreas, Desmond G. Higgins, and Cédric Notredame. 2008. "R-Coffee: A Method for Multiple Alignment of Non-Coding RNA." *Nucleic Acids Research* 36 (9): e52.

Woese, C. R. 1961. "A Nucleotide Triplet Code for Amino Acids." *Biochemical and Biophysical Research Communications* 5 (June): 88–93.

Woese, C. R., and G. E. Fox. 1977. "Phylogenetic Structure of the

Prokaryotic Domain: The Primary Kingdoms." *Proceedings of the National Academy of Sciences of the United States of America* 74 (11): 5088–90.

Wong, Karen M., Marc A. Suchard, and John P. Huelsenbeck. 2008. "Alignment Uncertainty and Genomic Analysis." *Science* 319 (5862): 473–76.

Yamada, Kazunori D., Kentaro Tomii, and Kazutaka Katoh. 2016. "Application of the MAFFT Sequence Alignment Program to Large Data—reexamination of the Usefulness of Chained Guide Trees." *Bioinformatics* 32 (21): 3246–51.

Zuckerkandl, E., and L. Pauling. 1965. "Molecules as Documents of Evolutionary History." *Journal of Theoretical Biology* 8 (2): 357–66.