Bioinformatics Research in Whole Genome Duplication

Julian M. Catchen

June 6, 2007

1 Introduction

Evolution advances the fitness of an organism through a series of small, incremental changes over long periods of time. The efficacy of these changes are constantly being tested against the constraints of natural selection – positive changes become fixed within the species, negative changes are lost through lack of reproduction. In contrast to this slow, stochastic process, the theory of whole genome duplication proposes a means for rapid evolutionary change and diversification. While normally a newly-conceived organism receives half of its chromosomes from one parent and half from the other, with a whole genome duplication the organism is given a full set of chromosomes from each parent – doubling the number of genes and shielding them from the effects of natural selection. The theory of whole genome duplication was first proposed in the early 1970's and was considered controversial until recently. Now, however, with the availability of a number of fully sequenced genomes, these duplications are known to have occurred a number of times in evolutionary history in a variety of different organisms, from the single-celled paramecium, to the salmon. Indeed, the ancient ancestors of humans are thought to have experienced two full duplications in their history and the teleost fish are thought to have experienced a third. The teleosts, which include the salmon, contain more species than any other vertebrates in existence today; one explanation for this explosion of diversity is the occurrence of three, incredibly rare, whole genome duplication events.

The earliest proposition for the theory of whole genome duplication used no genetic data at all; it simply was not available for comparison. Instead, it relied on a measurement of the physical size of an organism's chromosomes. The first evidence in favor of the theory began to accumulate in the second half of the 1990's as key genes were sequenced and mapped in the genome one at a time. The opportunity to collect enough sequence data to provide comprehensive evidence for the theory came with the arrival of the human genome project – beginning in the early 1990's, hitting its stride at the turn of the century, and completing a final draft of the sequence in 2003 [9]. The pace has only increased since then: one of the newest pieces of equipment, a 454 Sequencer, can produce a rough draft of an entire genome in a matter of days [56].

It is within this environment, now saturated with genomic sequence data, that computational resources are poised to advance the theory of whole genome duplication. Once the signal for these ancient events has been uncovered in modern organisms, there are many questions we would like to answer regarding the theory. For example, doubling the number of genes would create a chaotic environment within the cell, does selection quickly reduce the population of genes, or does it split the functions of a gene between the old and new copy? After a duplication, at what rates do normal mutations accumulate, and how do those

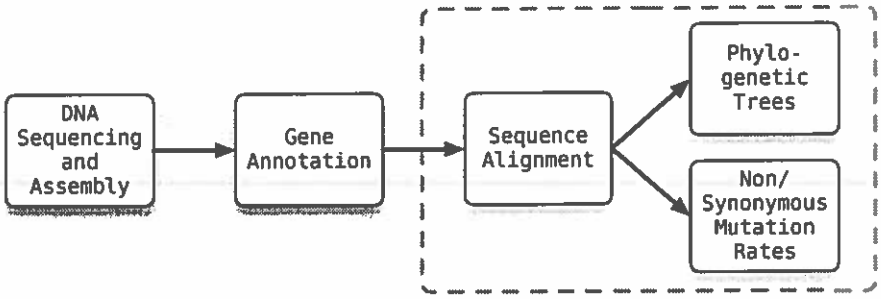


Figure 1: Several areas of focus within the field of Bioinformatics. Three of the key areas, enclosed in the gray box, are critical to research in the area of whole genome duplication and are the focus of this paper.

rates compare to the normal process of advancement? Does duplication cause speciation, and if so, do new species obtain novel features by following parallel evolutionary tracks, or does each species create the same novelty, through different mechanisms?

Several areas of focus within the field of Bioinformatics apply to the study of whole genome duplication and the questions posed above. These areas and the dependencies between them are illustrated in figure 1. DNA sequence data must first be obtained from the organism of interest. In the case of genome-wide sequencing, DNA is "read" in small, overlapping segments that must be assembled into the proper order by matching the overlapping ends together. Next, software is used to detect the location of genes within the DNA sequence as well as features within those genes. After these analyses have been completed, the full complement of an organism's genes is available for study. Alignment algorithms allow us to find genes that are biologically related to one another – both within an organism and between organisms. Once gene relationships have been annotated, phylogenetic trees can be built from them to infer the evolutionary relationships between genes, and, by identifying synonymous and nonsynonymous mutational differences between related genes we can infer the the effects of natural selection on those genes.

This paper will focus on three areas of research in the field of Bioinformatics including the methods, algorithms, and techniques that are critical to the study of whole genome duplication. These areas are highlighted by the gray box in figure 1:

- Alignment is a process of comparing the nucleotide or amino acid sequences between different organisms, or within a single organism, in order to identify conserved regions. It is the basis of every major informatics analysis and for that reason this paper will examine the relevant developments of Computer Science, Statistics, and Biology that have been integrated to address the computational complexity of the alignment problem. Specifically we will look at optimal sequence alignment, the statistical significance of alignments, substitution matrices and multiple sequence alignment.

- Phylogenetics is a set of methods used to determine evolutionary relationships between different organisms based on a tree representation. The tree, from its root out to the leaves, describes a precise ordering of speciation, from the ancient ancestral organism, to its modern-day descendants. Building phylogenetic trees from groups of related genes allows us to infer the ancestral changes in those genes – when novel features

appeared, how often they appeared, and perhaps when they were lost – by examining the present state of a gene in multiple organisms and using evolutionary models along with powerful statistics to work backwards and infer the ancestral states of a particular gene or gene family. This paper will focus on the four major tree-building methods: parsimony, distance, maximum likelihood, and Bayesian.

- The study of nonsynonymous and synonymous mutation rates allows us to infer the rates at which a set of related genes are changing. We rely on the neutral theory of evolution, also known as genetic drift, which describes the effects of random mutation and the fixation or loss of those mutations in a population, to determine if a particular gene is under the influence of positive or negative selection. Understanding these forces allows us to identify the phenomenon that occur after a genome has been duplicated. This paper will examine the neutral theory along with measurements for nonsynonymous and synonymous mutations.

- Finally, this paper will examine genome duplications, covering the theory behind them in detail as well as a review of past and current empirical evidence supporting them. To best understand the evidence in support or against whole genome duplication it is essential to understand the methods and algorithms used to generate the raw data and to understand the biological inferences drawn from that data; for that reason, the topic of whole genome duplication is presented last.

Section 1.1 examines some basic concepts in biology for computer scientists unfamiliar with or needing review of the fundamental processes involved in DNA transcription and translation. It will also introduce some basic vocabulary that will be needed later in the paper. Sections 2-5 address the fundamental results in the four areas described above: sequence alignment, phylogenetics, nonsynonymous/synonymous mutation rates, and whole genome duplication. Finally, section 6 concludes with a high-level look at the state of the art in whole genome duplication and discusses several open research questions in this area.

1.1 Biological Foundations for Computer Scientists

In 1965, Emile Zuckerkandl and Linus Pauling postulated which molecules were relevant to an organism's evolutionary history [83]. They wanted to classify what types of molecules could be used as a basis to compare organisms to one another and, through those comparisons, build a universal phylogenetic tree that described their evolution. Such a tree would reconstruct the lines of descent that resulted in the organisms that exist today (phylogenetic trees are described in detail in section 3).

According to Zuckerkandl and Pauling, molecules that occur in living matter can be classified into three categories relative to the amount of information about the organism that the molecule contains. Semantophoretic molecules (semantides) are molecules that directly carry genetic information or a transcript of that information. DNA molecules are the primary example of this class of molecules which also includes messenger RNA (mRNA) molecules and the polypeptides that are the products of DNA and mRNA molecules. The second class of information carriers are episemantic molecules and include any molecules synthesized under the control of the polypeptides created by the semantides. All molecules built by enzymes, which are proteins synthesized by genes, belong to this class. Finally, asemantic molecules represent a class of molecules that are not produced by the organism. These molecules may be used and modified by the organism and may be used as a source for episemantic molecules. Vitamins are an example of this class of molecules.

Asemantic molecules are not useful in terms of building relationships between different organisms because the presence of such compounds in an organism can only imply that the organism possesses the proper episemantic molecules to utilize the molecule. Episemantic molecules, in turn, can only tell us that the organisms have similar active sites in their semantides, or possibly entirely separate pathways to produce the same product.

For these reasons, Zuckerkandl and Pauling assert that "the most rational, universal, and informative molecular phylogeny will be built on semantophoretic molecules alone." Further, by comparing the polypeptide products of related organisms, we should be able to determine the approximate time a polypeptide chain came into existence, the probable amino acid sequence of the last common ancestor, and the lines of descent from which various changes occurred. They also state that additional information can be gathered by examining the three-dimensional structure of a protein and by examining its direct genetic code (DNA nucleotides). Almost every bioinformatic analysis starts with the comparison of homologous semantophoretic molecules.

1.2 Important Biological Concepts

Although an extensive treatment of DNA and all of the processes involved in its transcription and translation is beyond the scope of this work (see [51], [61], and [57] for an introduction), we will briefly describe some biological concepts as they relate to the topics in this position paper.

Every living organism contains a linearly arranged set of information that describes a series of genes [83]. These genes describe how to build and execute all the systems that make up the organism, from describing the organism's body plan to the regulation of the number of white blood cells for the immune system. This deoxyribonucleic acid (DNA), which is a semantide, is present in every cell of every organism from single-celled bacteria to complex organisms with multiple, cooperating tissue types and internal organs such as mammals.

DNA is composed of four types of nucleotides, which are known by their bases adenine (A), cytosine (C), guanine (G), and thymine (T). These bases can be classified into two categories based on their chemistry, the purines and the pyrimidines, that naturally pair with one another - the purine adenine with the pyrmidine thymine as well as the purine guanine with the pyrimidine cytosine. DNA is composed of two strands of these nucleotides, complementary to one another and arranged as a double-helix. Due to this complementary nature, if given one strand of the DNA, the other strand may be re-constructed from it.

The DNA strands encode a series of genes or functional units a portion of which are protein-coding genes. The beginning and ending of each gene is marked by a particular set of nucleotides and, internally, each gene contains one or more exons and introns. Exons and introns are differentiated by the fact that the code specified within an exon will become part of the final protein, whereas the code contained within an intron will be spliced out of the sequence before the final protein is completed. Introns are thought to serve a regulatory role in the production of the protein. Areas immediately preceding a gene, known as promoter regions, regulate the circumstances under which that gene is read (see figure 2). Interestingly, only a very small fraction of an organism's genome contains code for functional genes, for humans, it is only 3%. The remaining 97%, known as nongenic DNA, was popularly described as "junk" DNA for a time and is not well understood. Some regions of nongenic DNA are known to be genes that have been rendered non-functional by mutations (commonly referred to as pseudogenes), while other portions are thought to serve a regulatory

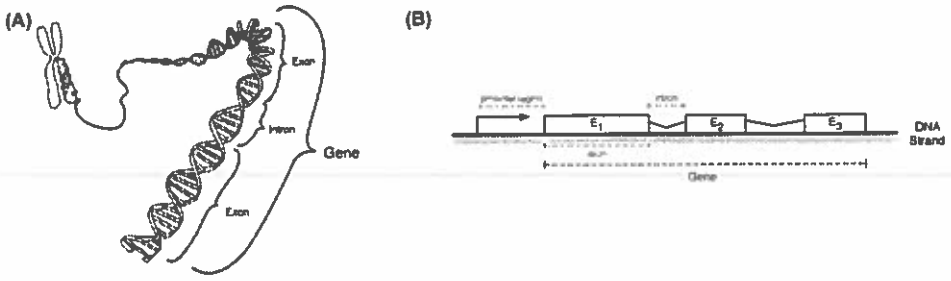


Figure 2: Two illustrations of a gene. (A) This physical representation shows how the gene is arranged, along with its introns and exons, within the DNA double helix and where it is stored on the chromosome. Illustration from [49]. (B) This representation shows the basic layout of a gene on a strand of DNA along with its functional units. Pictured, from left to right is a promoter region, followed by three exons (E_1, E_2, and E_3), separated by two introns (angled lines).

purpose [51].

In order to create a protein, the internal machinery of the cell first splits the DNA strands and reads the nucleotides belonging to a particular gene. This process is known as transcription and it produces a complementary strand of RNA called the primary transcript. After this initial reading, the primary transcript contains a faithful copy of the DNA including exons as well as introns. However, the primary transcript is generally further processed by splicing out the introns to make messenger RNA (mRNA) and possibly some of the exons to make splice variants.

Within a coding section, each group of three nucleotides, known as a codon, specifies a particular amino acid. A gene encodes for a series of amino acids that are combined to create the protein product of the gene. Since there are four different nucleotide bases, it is possible to encode $4^3 = 64$ amino acids. However, only twenty different types of amino acids are used in the formation of proteins and, therefore, multiple codons can specify a single amino acid. For this reason, the genetic code is referred to as degenerate [83, 51].

Once processing of the mRNA is complete, it moves from the cell nucleus into the cytoplasm where cellular ribosomes attach to it and begin the process of translation. During this process, the codons that make up the mRNA are read, and the corresponding amino acids are fetched and attached to one another creating a chain of polypeptides. As this chain is assembled, the polypeptides fold into a final, three-dimensional protein that, when complete, is then utilized by the organism in some functional way. For example, the protein may act as a signaling protein triggering additional proteins to be synthesized or it may be involved in catalyzing another chemical reaction within the organism. This process is illustrated in figure 3.

Organisms that have a relatively recent common ancestor share significant portions of their DNA including many protein-coding genes. Many times, a whole gene, portions of a gene, or even whole segments of a chromosome are conserved between organisms. However, because of mutations and other evolutionary changes, the code is rarely identical in different species, or even in different individuals of the same species. Enumerating these differences allows us to make many inferences about the organisms. Due to the degeneracy of the genetic code, comparison of segments of the genetic code translated into amino acids is often more

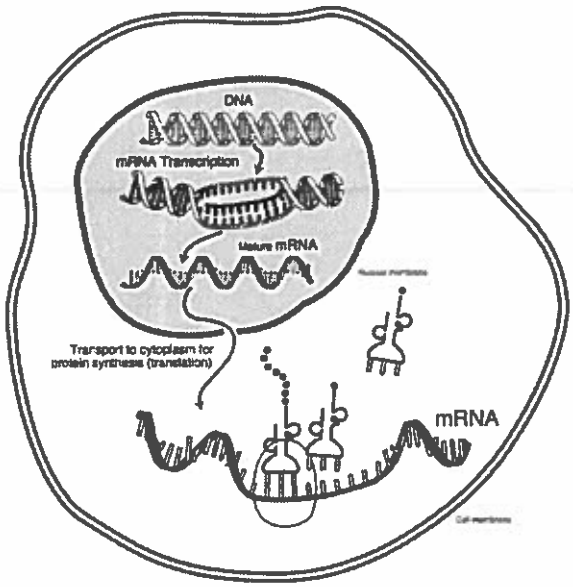


Figure 3: An illustration of the transcription and translation process [50].

forgiving than those performed with nucleotides since many nucleotide mutations do not alter the resulting amino acid. For this reason, amino acid translations are often used when comparing distantly related sequences.

Figure 4 describes several common relationships among genes; identification of these relationships is often used as a basis for inference. For example, zebrafish and pufferfish diverged as two distinct species much more recently then the ancestral line that diverged to become the ancestors of fish and humans. Finding a gene that is present in both the zebrafish and the pufferfish, but absent in humans implies one of two evolutionary histories: either the gene appeared in the fish lineage after humans diverged from fish, or, the gene was present in the last common ancestor of fish and humans and was lost in the human lineage. Genes that are related by a common ancestor in the past are referred to as **homologs**. A gene that is present in both humans and zebrafish and was a single gene in their last common ancestor is known as an **ortholog** (figure 4(B)). Sometimes, genes, chromosomal regions, or whole chromosomes duplicate in an organism resulting in the doubling of all genes in the duplicated region. In rare cases, whole genomes have duplicated in a single event (see section 5 for more details). These genes are known as **paralogs** of one another (figure 4(A)). Thus, orthologs are two genes that arise from a speciation event, and paralogs are two genes that arose from a gene duplication event within a lineage. When a set of paralogs in one organism, and the ortholog of those paralogs in another organism are still related to a single gene in the last common ancestor they are known as **co-orthologs** (figure 4(C)). Co-ortholog gene relationships are most commonly found when comparing genes from an organism that has experienced a full genome duplication with a genome that has not, for example, comparing humans and zebrafish. Co-orthologs, paralogs, and orthologs are all more specific cases of homologs.

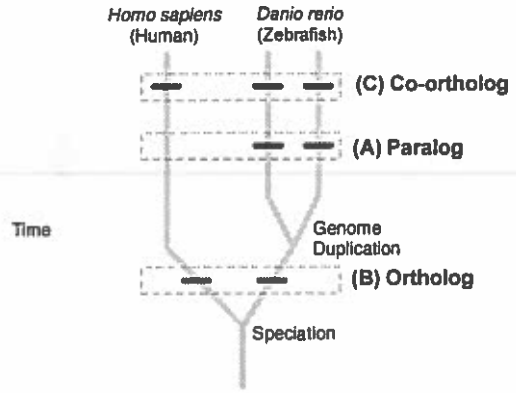


Figure 4: Genetic Relationships. The above tree represents the line of descent of humans and zebrafish for a hypothetical chromosome, showing their last common ancestor, a speciation event, and a genome duplication event (causing the single zebrafish chromosome to become two chromosomes, doubling all the genes on the original chromosome) in the zebrafish lineage. (A) Paralogs: genes within the same genome descended from a single gene that was duplicated. (B) Orthologs: genes in different genomes descended from a single gene in the last common ancestor of those genomes. (C) Co-Orthologs: a set of paralog genes, along with a single, non-duplicated ortholog of those genes in another genome all descended from a single gene in the last common ancestor. (A-C) Each set of genes are homologs of one another.

2 Sequence Alignment and its Significance

As asserted by Zuckerkandl, and reiterated by Russell Doolittle in [14], it is simpler to duplicate and modify an existing genetic program defining the composition of a protein than to assemble a set of proper amino acids randomly. That is, novel function develops in an organism over long periods of time through small modifications to their genetic material. The proliferation and divergence of that genetic material through a series of organisms leaves a signal that can be revealed by comparative analysis of sequence data.

In the case of a whole genome duplication, the amount of genetic material is doubled in single, swift event, while the evolutionary implications resulting from that event span millions of years. As an example, figure 5 shows the signal of a duplication event in the pufferfish – revealed through the identification of paralogous and orthologous genes through sequence alignment. The left plot of the figure shows the results of taking each of the 1,100 protein-coding genes on the pufferfish's third chromosome and searching among all 28,000 genes in the pufferfish genome for paralogous genes. These genes are represented in the plot by a line of light gray dots while matching paralogs are shown as red points. What should be evident is that there is a clear pattern of duplicated genes present on the pufferfish's second chromosome. To verify that these genes are indeed duplicates of one another, they were compared against an organism that has not experienced a whole genome duplication since the species diverged evolutionarily. In this case, each set of pufferfish paralogs was compared against the 32,000 protein-coding genes in the human genome in search of co-orthologs. Again, a clear signal is present with the co-orthologous human genes appearing along a single, broken human chromosome which is composed of half of human chromosome

Figure 5: Data, generated by the author, demonstrating a whole genome duplication in the pufferfish – a member of the teleosts. The left plot shows that genes on pufferfish chromosome 3 are duplicated on chromosome 2, while the right plot demonstrates that the corresponding human co-orthologs were not duplicated, but instead remain half on chromosome 2 and half on chromosome 17. The points in the plots were enhanced due to the low resolution of the images.

two and half of human chromosome seventeen. From these two plots we can infer that the last common ancestor of humans and pufferfish possessed a single chromosome that has since been duplicated in the pufferfish, and broken into two pieces in humans.

While these two plots appear to be simple, they are based on thousands of comparisons of genes, where each of those comparisons relies on sequence alignment – fuzzy matching of nucleotides or amino acids that can account for mutations and changes that have accumulated over millions of years. Those alignments, in turn, rely on a solid foundation of statistics to identify alignments that are significant, and those statistics rely on evolutionary models and empirical data to demonstrate which statistically important alignments are biologically relevant. Whatever underlies our methods of analysis, if we want to make inferences on a genome-wide scale our algorithms must be fast!

As this example demonstrates, sequence alignment can be used to establish relationships among genes, and those relationships can be used to infer historical evolutionary events. Moreover, almost every major Bioinformatics analysis depends on alignments as a starting point – both phylogenetic tree building and the estimation of synonymous and nonsynonymous mutation rates require alignments before their evolutionary models and powerful statistical techniques can be brought to bear. In the remainder of this section we will examine these issues by looking at each of the major areas of sequence alignment including fundamental algorithms to align sequences, the statistics underlying those algorithms, a case study looking at the most widely-used alignment program, multiple alignment algorithms, and a brief look at areas of on-going work in the field.

2.1 Mathematically Optimal Sequence Alignment

In 1970, Saul Needleman and Christian Wunsch published the first general method to compare two sequences using a computer-based algorithm. They defined the maximal match between two protein sequences as the largest number of amino acids of one protein that can be matched with those of a second protein, while preserving order and allowing for possible insertions or deletions (indels) [58].

In 1974, Peter Sellers introduced another method for aligning sequences. Sellers' method was presented in a mathematically rigorous way and, instead of trying to **maximize** the

number of similar amino acids, Sellers's algorithm sought to **minimize** the evolutionary distance between a pair of sequences [66]. The evolutionary distance, or edit distance, is the number of changes that must be made to one sequence (e.g. insertions, deletions, and substitutions) to change it into the second sequence. The greater the number of changes, the greater the amount of evolutionary time that has passed since the organisms diverged. Using evolutionary distance as a metric allows us to reason about sequences using the principle of parsimony. Although parsimony is discussed in depth in section 3, briefly, it asserts the idea that the smallest number of changes to the sequences is likely to be the best explanation of their evolution.

Several years later Temple Smith and Michael Waterman proved the original Needleman-Wunsch algorithm and showed that the set of alignments achieving maximum sequence identity for the Needleman-Wunsch algorithm was equal to the set of alignments achieving minimum evolutionary distance for Sellers's algorithm [68].

Although Needleman and Wunsch did not describe it as such, their algorithm is a version of dynamic programming [48, 57, 15]. Their original formulation was not as efficient as current methods and it operated somewhat differently; for example, it aligned the sequences from the terminal residues to the beginning. Instead, we will describe a more mainstream and recent version [17, 15, 57].

A naive method to determine the best alignment between two sequences would be to simply try all possible combinations of the letters in the two sequences and then determine which one had the best score. Even with sequences of modest length it is easy to see that the number of possible alignments would be very large. Additionally, with each individual alignment, we will be repeating work many times as we align common subsequences. Dynamic programming provides a solution to this problem by relying on the idea that we can construct an alignment between two sequences by building it up from alignments of smaller subsequences. To avoid duplicated work the dynamic programming technique records the aligned subsequences so they can be reused without having to recalculate them later in the alignment process. Finally, dynamic programming provides an optimal alignment solution to the problem. This does not imply that the alignment is biologically optimal, rather, given a particular scoring system, dynamic programming will find the alignment with the best score.

To determine the optimal alignment of two sequences we place one sequence along each axis of a two-dimensional matrix and paths through the matrix define possible alignments between the sequences. The value contained in cell (i, j) represents the cost of aligning the first i characters of one sequence along with the first j characters of another sequence. We calculate the cost for each cell in the matrix and to determine the optimal alignment we simply find the path through the matrix that minimizes this cost.

One such matrix, which was generated with actual sequence data for this paper, is illustrated in figure 6 (A). In this example, we are aligning a portion of the human engrailed gene (*EN1*) against the zebrafish ortholog (*eng1a*) [6]. The human gene runs along the X axis and has a length of m. The zebrafish gene runs along the Y axis and has a length n. To determine the value of each cell in the matrix we define a recursive relation that describes the value of the cell (i, j), where $0 < i < n$ and $0 < j < m$, in terms of its three neighboring cells:

$$s(i,j) = max \begin{cases} s(i-1, j-1) + \sigma_{ij}, \\ s(i-1, j) + \gamma, \\ s(i, j-1) + \gamma. \end{cases}$$

This formula describes three options: 1) extend an existing alignment, either because the

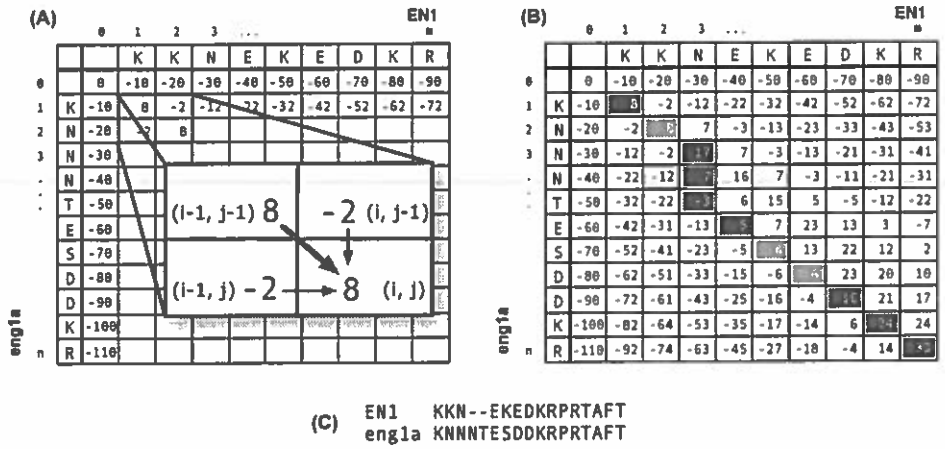


Figure 6: An illustration of the global alignment algorithm using the human engrailed gene (*EN1*) and the zebrafish ortholog (*engla*). (A) Calculating the value of cell (i, j) based on its three neighbors: $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$. We used a gap penalty of $\gamma = -10$ and a BLOSUM matrix (defined in section 2.4) as our scoring function. In this case, the scoring function $\sigma_{i,j}$ specifies the cost to align the amino acids K and N as 0. Therefore, we select the maximum of the three choices: $s(i-1, j-1)+\sigma_{ij} = 8+0$; $s(i-1, j) + \gamma = -2 + -10$; $s(i, j-1) + \gamma = -2 + -10$. (B) The final alignment as represented as a path of red and blue cells through the dynamic programming matrix. Red cells represent an exact match, light red a mismatch, and blue cells represent a gap. (C) The final alignment.

residues match or because a substitution has occurred which is more acceptable than inserting a gap, 2) add a gap in the first sequence, or, 3) add a gap in the second sequence. To determine the degree of similarity between two residues we use a scoring function, σ_{ij}, which is defined in terms of the probability that residue at position i of the first sequence is related to the residue at position j of the second sequence. A scoring function can be as simple as returning a positive integer for two matching residues and a negative integer for two mismatched residues. Or, more commonly, the scoring function will return a large integer for a match, and a slightly smaller integer for a mismatched, but biologically similar residue. In general, the scoring system is based on an evolutionary model or empirical evidence and several popular scoring systems are detailed in section 2.4. In addition to the scoring function, we also have a gap penalty γ, and if the two residues are so unrelated that the value of the gap penalty is larger than the score from σ, then the algorithm will choose to insert a gap into the alignment (in this example we use a gap penalty of -10).

After initializing the top row and column in the matrix we iterate over each cell and calculate the value of the cell based on the scoring function and the gap penalty. We select the option that provides for the maximal value and we record which option we chose (left, diagonal, or up). Because we are recording these choices, we can construct a path from any position in the matrix back to the originating cell. Pathways that travel along a diagonal in the matrix represent portions of the two sequences that can be aligned without gaps (but possibly with substitutions), while horizontal and vertical pathways represent insertions and deletions in the sequences.

Once the matrix is fully calculated, we determine the optimum path through the matrix

by starting at the terminal cell (n, m) and retracing our steps through the matrix until we arrive back at the origin of the two sequences. This trace is shown in figure 6 (B) and the final alignment can be seen in figure 6 (C). Since we have to determine a value for each of the n x m cells, where each cell requires constant work, the algorithm executes in time proportional to the product of the sequence lengths, $O(nm)$.

2.2 Statistical and Biological Significance of Alignments

In the section above we described how to determine a mathematically optimal alignment of two sequences. As an example, we aligned portions of the human and zebrafish engrailed gene. As shown in figure 6 (B), the result of the exercise was a score for the alignment: 33. It is clear where this score comes from – it is the sum of the scores σ_{ij} that are incurred when aligning the letter at position i of the first sequence with the letter at position j of the second sequence, in addition to any gap penalties. Beyond knowing that the number 33 is the highest score we could obtain aligning that portion of the engrailed gene, what is its larger significance? Is it more significant than if we had aligned two randomly generated sequences? In general, how do we identify genes that share a common history and how do we differentiate authentic relationships between genes from those occurring by chance?

In the following sections we will examine these questions. We will first look at some basic considerations of sequence alignment and some simple methods to measure significance. After discussing the shortcomings of these methods we will describe a general scoring method that is used widely today. We will focus on choosing an alphabet and scoring system, including how to generate the major substitution matrices, and how to apply the system to single sequences and finally sequence alignments.

2.3 Influencing Factors in Sequence Alignment and Naive Statistical Significance Measures

We start our discussion of statistical significance with an examination of some basic properties of genetic sequences. If we assume for the moment that amino acids occur throughout the genome with equal frequency then two unrelated protein sequences that are the same length can be expected to have an average of five percent identity by chance alone since there are twenty amino acids. Likewise, under the same conditions, two nucleotide sequences being compared would be expected to have a twenty-five percent identity by random chance since there are only four nucleotides. With a large number of sequence comparisons the distribution of sequence identity should approximate a normal curve with a mean at five percent and twenty-five percent respectively [14]. Considering only amino acids, 95% of the comparisons made should fall within two standard deviations of either side of the mean of the normally distributed data. However, depending on the length of the sequences, the standard deviation itself is going to vary. With shorter sequences the standard deviation will be larger and will decrease as the length of the sequences increases (since longer sequences provide more data causing the sample mean to move closer to the true mean). Doolittle provides the example of comparing sequences 50 residues long and 200 residues long. In the former case, 95% of the comparisons will have between zero and eleven percent identity. In the latter case, the expected percent identity falls to an interval between zero and nine percent as the standard deviation gets smaller. In terms of using percent identity as a measure of statistical significance, we must account for the lengths of the sequences [14].

Because the process of change that occurs to genes over time often involves insertions and deletions of nucleotides, expecting our sequences to always be the same length is not realistic. If we allow the sequences being compared to be shifted left or right with respect to one another, perhaps because one of the proteins lost one of its ends, we see the mean of our distribution of percent identity change. If we allow one sequence to be shifted five residues the mean of our distribution increases from five to eight percent (since we now take the maximum of two comparisons between a pair of sequences) and 95% of our sequence comparisons will have between four and twelve percent identity. Finally, if we now allow gaps to appear in the sequences, to allow for insertions and deletions in the middle of the sequences, then two sufficiently long, but completely unrelated sequences can show artificially high levels of identity. Clearly, to demonstrate statistical and biological significance, sequence comparisons must account for sequence length and gap placement [14].

Based on these facts, Doolittle proposed one of the earliest methods for determining the relevance of a number of similar sequences. The process starts by first searching a database of available sequences with very liberal search criteria in order to generate a list of sequences possibly related to the query sequence. Next, for each candidate match an optimal alignment is calculated using a gap penalty in the scoring routine to limit gap placement. A series of random sequences of similar length and composition to the originals are then generated and aligned in the same way as the actual sequences were. The score from each alignment of the random pairs of sequences is aggregated and the mean and standard deviation are calculated providing a distribution that can be used to compare actual sequence alignments against. Any sequence alignment score that lies at least three standard deviations above the mean percent identity of the randomized distribution would safely indicate a biologically significant relationship between the two sequences [14].

In 1983, Fitch and Smith confirmed and expanded on Doolittle's conclusions. They compared chicken hemoglobin sequences to explore gap parameters and statistical significance [27]. Their results indicate that not only should gaps be weighted, but they should be specified with two parameters: a gap opening penalty as well as a gap expansion penalty. The authors found that the gap penalty must increase monotonically with the length of the gap, and, a gap of five residues, for example, must be considered a single gap, as opposed to five separate gaps. Additionally, Fitch and Smith reiterated the difference between statistical significance and biological similarity, echoing the need to create a measure of significance using a randomization and standard deviation method similar to what Doolittle proposed.

Lipman, Smith and colleagues focused further on the question of statistical and biological significance versus chance relationships by examining the composition of the randomized sequences used to compare against actual alignments [54]. Until this point we have considered the amino acid or nucleotide components of genetic sequences to occur at equal rates throughout the sequences we have examined. Lipman proposed that since actual genetic sequences are not randomly generated but are subject to functional and evolutionary constraints, these factors must be accounted for when generating random sequences to determine statistical significance. More specifically, the authors asserted that nearest neighbor frequencies as well as local fluctuations in base composition can greatly affect the similarity between randomly generated sequences. Nearest neighbor frequency refers to the idea that certain amino acids are likely to be found near other particular amino acids, such that if you create sequences totally randomly, you are likely to decrease the percent identity relative to non-random sequence data. Fluctuations in local base composition refers to the distribution of the four nucleotides in sequence data and the fact that different segments of genomic sequence are going to have different rates of occurrence of the four nucleotides

depending on what types of biological structures the sequence is encoding.

To illustrate these factors, Lipman and colleagues took 100 vertebrate nucleotide sequences randomly chosen from GenBank and selected ten of them to be query sequences. They then aligned each of the ten query sequences against all 100 GenBank sequences and used the scores of the resulting 1,000 comparisons to approximate the true distribution of the sequence comparison scores. Next, three sets of randomized sequences were generated from the GenBank sequences and the same analysis was repeated. Each set of randomized sequences preserved a particular characteristic of the original sequences: overall base composition, nearest neighbor frequency, or local base composition, respectively. The authors found that the three sets of randomized sequences produced significantly different percent identity distributions. After further analysis, comparing four real pairs of sequences against the random models, they found that different models could account for some of the pairs, but none of the models could account for them all. For example, the percent identity of one of the real alignments may have been considered statistically significant when compared against the set of random sequences where overall base composition was preserved but not under the model where nearest-neighbor frequencies were preserved. These results demonstrated the difference between abstract statistical significance and true biological significance when trying to determine the relationship between different genetic sequences. Although Lipman and colleagues suggested producing several variations of randomized sequences to determine the significance of all sequence comparisons a statistical basis for general comparison was published shortly after.

In general, the early work of Doolittle, Fitch, and Lipman established some basic statistical limits to working with biological sequence data. Namely, they started with the assumption that nucleotides or amino acids occur randomly in sequence data and that the scores resulting from alignments of that data are distributed normally. Through experimentation they found that the patterns in which nucleotides and amino acids occur, along with the length of the sequences affect the statistical significance of alignments. Additionally, they determined that the insertion of gaps, or the shifting of sequences – allowing us to improve alignments by selecting the maximum scoring of the shifted alignments – also affects the statistical significance. In the next section we will look at some of the problems with these ad-hoc approaches, and following that we will examine a formal statistical framework that describes the distribution of sequence alignment scores, superseding this early work.

2.3.1 Shortcomings of Ad-Hoc Scoring Methods

As described above, a naive approach to determining the statistical significance of a sequence alignment involves comparing the score of an alignment against the mean of a distribution of similar sequence alignments where the underlying sequence data has been randomized in a biologically meaningful way. The statistical significance of the alignment is reported as the number of standard deviations it falls above the mean of the randomized distribution. However, as Altschul and colleagues describe in [2], there are several problems with this approach.

First, this method relies on a distribution of alignment scores with which to compare alignments against. Randomizing a set of sequences and aligning them is a way to approximate the true underlying distribution of alignment scores, however, for this approximation to work the randomized sequences must belong to the same distribution – an assumption we can not make with the sequences from any particular collection. As Lipman demonstrated, variation in residue composition among sequences can yield different alignment score distri-

butions. Further, we also must take the length of sequences into account, as the longer a pair of sequences, the greater the alignment score expected by chance.

Second, up until this point, we have assumed that optimal alignment scores are normally distributed. This would be the case if finding an optimal alignment involved summing many independent subalignment scores. However, when we are determining the best alignment score we are instead seeking the **maximum** of many independent subalignment scores and this results in an extreme value distribution (EVD). One way to visualize the difference is to think about a dynamic programming matrix used to align two sequences. If, once the matrix is filled in, all paths were equally likely, we would simply choose a path and sum the scores along it. However, all paths are not equally likely, and instead we repeatedly choose the maximum of the scores to find an optimal path through the matrix. This difference in the way subalignment scores are selected results in a distribution that is positively skewed and has a tail that extends much farther out to the right from the mean than a normal distribution does (see figure 7). So, if a normal distribution is assumed, the significance of an alignment will be greatly exaggerated (the EVD will be discussed further below).

2.3.2 General Scoring Statistics

As an alternative to determining the statistical significance of an alignment by measuring the number of standard deviations the alignment score falls from the mean, Samuel Karlin and Stephen Altschul developed a general method for determining the statistical and biological significance of individual sequences and sequence alignments [42]. We will examine this general method first by describing how to choose an alphabet and define a scoring system, and then by examining how Karlin and Altschul's method applies to single sequences followed by extending it to pairs of aligned sequences.

Before we begin, we must make a distinction between global alignment and local alignments. In the former case, the goal is to optimize the overall alignment of two sequences which may include long regions of low similarity. Local alignment, on the other hand, seeks only to align those portions of sequences that are relatively conserved. A global alignment will always produce a single definitive alignment between two sequences while a local alignment may produce several significant subalignments from the same pair of sequences. The techniques described below have only been shown to be valid for local alignments – the similarity statistics do not necessarily hold when the alignment must include long stretches of low similarity.

We saw previously that randomizing and aligning a subset of sequences in order to approximate the distribution of their alignment scores was done because the actual distribution of the alignment scores was not known. The power in Karlin and Altschul's general method is that by defining a scoring system, which is based on a chosen alphabet, the distribution of alignment scores can be analytically determined, allowing statistical significance of alignment scores to be measured without making assumptions about an unknown underlying distribution. As a prelude to sequence alignments we will first look at finding features in single sequences. The alphabet and scoring system, which we will describe here, are common to both tasks.

Karlin and Altschul's method begins first by declaring an alphabet to use, where the elements of the alphabet represent the components of a sequence to which we wish to assign a score. For example, if we want to identify the subsequence within a gene responsible for loops or coils in the secondary structure of a folded protein, then we are going to want to use an amino acid alphabet. If instead we want to identify repetitive elements of a sequence we may

use a nucleotide alphabet. In general, an alphabet with r elements is specified as $a_1, a_2, ..., a_r$ and a set of corresponding scores for each of the elements is specified as $s_1, s_2, ..., s_r$. As we have seen, a number of different alphabets are possible, including nucleotides, in which case $r = 4$ and the alphabet is composed of $a_1 = A, a_2 = C, a_3 = T$, and $a_4 = G$; codons, with $r = 61$ (64 possible codons minus the three nonsense/stop codons); and the standard amino acids, with $r = 20$. Other potentially useful alphabets may focus on characteristics such as the chemical nature of the nucleotides where $r = 2$ with purines and pyrimidines as the components.

The scores themselves can be based on several attributes depending on what types of features in a sequence or between sequences we want to recognize. For example, scores may be based on the charge of amino acids or the chemical similarity of different amino acids. While we will discuss scores more in the context of aligning pairs of sequences, when evaluating single sequences, most commonly, scores will be based on a set of target frequencies. The idea here is that a feature can be identified when the letters of the alphabet that define that feature occur more frequently than they would when compared against their normal background frequency (the frequency that they occur in sequences without the feature). We can express this in the following way: the letters of the alphabet that define the feature of interest occur with probabilities $q_1, q_2, ..., q_r$ and the background frequencies of the letters are $p_1, p_2, ..., p_r$. The score for this feature may then be expressed as a log-likelihood ratio: $\sigma_i = \log(q_i/p_i)$. In effect, the ratio compares the probability of an event occurring under two alternative hypotheses – the letters of the alphabet appeared because we have identified the feature, versus the letters having appeared simply because of chance (background frequency) [1]. So, if we wanted to identify a set of cell transmembrane regions where the protein glycine appears more often than in normal sequences, we simply tweak the target frequencies to favor glycine [42].

2.3.3 Maximal Segment Score and the Extreme Value Distribution

Given a scoring system that has been created to recognize certain features in a sequence, we can apply it to find the maximal segment score (MSS), the portion of a sequence of length n that has the highest aggregate score. This can be done by scanning the sequence with a window of fixed size and summing the scores of the residues within the window [42]. Once we have identified the MSS, we can determine if it is statistically significant by comparing the MSS against a distribution of maximal segment scores. These values are described by the extreme value distribution (EVD), which is defined by the cumulative distribution function:

$$P(S < x) = \exp[-e^{-\lambda(x-u)}]$$

[3] and includes two parameters, λ and u, which we must calculate. λ serves as a scaling factor, allowing different scoring systems to be compared to one another [1]. It is calculated first and is the unique positive solution to the following equation (for more information on λ see section 2.4):

$$\sum_{i=1}^{r} p_i e^{\lambda \sigma_i} = 1$$

where p_i is the probability of observing the ith letter of our alphabet and σ_i is the score of that letter. The second parameter, u, is called the characteristic value and can be thought of as the center of the distribution [3]. It is calculated as follows:

$$u = (\ln Kn)/\lambda$$

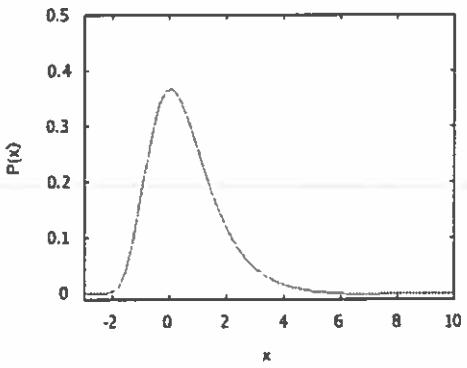


Figure 7: The probability density function for the extreme value distribution with $\lambda = 1$ and $u = 0$.

where n is the sequence length and K is a constant derived from λ and the expected values of σ (a full derivation is given in [42]).

Combining the cumulative distribution function for the EVD and the definition of u the distribution is finally given by:

$$P(MSS > x) = 1 - \exp\left(-Kne^{-\lambda x}\right)$$

and is shown in figure 7. The variable x, then, is a score; the equation gives the probability of finding a score above a given value of x and should decrease with increasing values of x. For example, if we set the equation P equal to 0.01, and solve for x, then any maximal segment score that is greater than the resulting value can be considered statistically significant at the 1% level [42].

2.3.4 Generalizing to Pairs of Sequences

The process of identifying maximal segment scores can be generalized for sequence comparison. There are two major changes that accompany this generalization; we have to shift our scoring system from one used to identify features in a single sequence to identifying shared regions between sequences and, second, we will now be searching for ungapped local alignments between the two sequences, which can be thought of as finding corresponding maximal segment scores in each of the sequences. This generalization adds some new considerations in our choice of alphabet. For example, if we want to compare two sequences at a fine level of detail, perhaps in search of paralogous gene relationships, we may want to use a nucleotide alphabet. If we want to compare distantly related sequences, perhaps in search of orthologous gene relationships, we may choose an alphabet of amino acids since individual nucleotide mutations may not change the resulting amino acid. If we want to be able to detect how radically a protein has changed we may choose an alphabet of codons since it allows us to detect all of the nucleotide changes, as well as which ones actually changed the resulting amino acid the codon describes.

Whereas previously we defined an alphabet and a set of probabilities for that alphabet, $p_1, p_2, ..., p_r$, we now add a second set of probabilities to represent the likelihood of letters in a second sequence, $p_1', p_2', ..., p_r'$ and we consider pairs of letters from our alphabet, a_i

and a_j, representing letters from the first and second sequence respectively. We will also need a new scoring scheme, one in which the target frequencies, q_{ij}, refer to the likelihood of letters a_i and a_j of our alphabet appearing in the same position of the sequences we are comparing. We again use a log-likelihood ratio to express the probability of the two letters appearing together because they are related versus them appearing by chance alone: $\sigma_{ij} = \log(q_{ij}/p_i p_j')$. There are several methods to determine the q and p values in our scoring scheme and we will detail two of them in section 2.4. Finally, the highest scoring ungapped alignment is now known as the maximum segment pair (MSP) [42].

Given two sequences of length m and n, that are roughly the same size, we can adapt our previous definition and again find the unique solution for λ, $\sum_{i=1}^{r} p_i p_j' e^{\lambda\sigma_{ij}} = 1$. The characteristic value, u, is now given by $u = (\ln Kmn)/\lambda$. The probability that the optimal alignment attains a score of at least x is finally given by:

$$P(MSP > x) = 1 - \exp(-Kmne^{-\lambda x})$$

[3].

These analytical solutions have only been proven for ungapped alignments. Altschul and Karlin have asserted, that according to empirical data the theory for determining statistical significance should extend to gapped alignments; however, the parameters λ and u cannot be calculated directly. Instead, they must be estimated by generating random sequence data of the same type and composition, and fitting the parameters to this data using numerical methods such as method of moments, maximum likelihood, or linear regression [3].

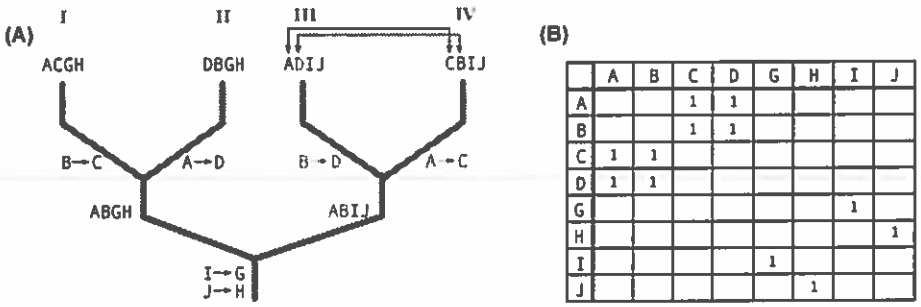
When comparing a single sequence to all the sequences in a database, the lengths of the two sequences, m and n must be adjusted. While m represents the length of the query, n must be set to the number of residues in the entire database to obtain an accurate upper bound on the number of distinct maximal segment pairs that the search is expected to produce [1].

2.3.5 Sum Statistics

Up until this point, we have only considered examining a single sequence for interesting patterns of composition or comparing an optimally aligned pair of sequences. When examining a single sequence, you may find several regions that all have high scores that you want to consider or, when examining a pair of sequences of sufficient length, it may be more useful to optimally align several portions of the sequences separately, in effect producing a type of gapped alignment (in section 2.5 we will look at BLAST which searches for similar sequences by performing a number of local alignments). In these cases, it can be useful to not only consider the maximum segment score, but instead to consider the top n scoring segments in a statistically meaningful way.

There are two primary methods to consider the top n scoring segments together: the first takes advantage of the fact that the number of high-scoring segments expected to have a score greater than x by chance is approximated by a Poisson distribution with parameter $e^{-\lambda(x-u)}$. For example, if we align two long sequences and find three high-scoring segments, with scores 40, 45, and 50, we can calculate the probability that at least three maximum scoring pairs, all with scores above 40, would appear by chance [42, 2]. The main drawback of this approach is that it relies on the lowest of the top n scores considered.

A second method sums the scores of the top n segment pairs. In this case, the distribution describing the sum of the n highest scores, T_n, has been derived numerically and it can be used in the same way as the EVD. The T_n distribution describes the probability of observing



	A	B	C	D	G	H	I	J
A			1	1				
B			1	1				
C	1	1						
D	1	1						
G							1	
H								1
I					1			
J						1		

Figure 8: PAM substitution matrices are built by constructing a phylogenetic tree of related protein sequences and counting the number of point mutations in the inferred ancestors. (A) Phylogenetic tree composed from four homologous proteins (labeled I - IV) and with the six point mutations that have occurred on the internal nodes of the tree labeled. Two of these point mutations are colored and marked with arrows to show exactly which nucleotides were involved in the determination. (B) Since the point mutations are considered reflexive, 12 entries representing the mutations are recorded in the table. This figure is based on [11].

the sum of n scores above a given value of x. For a more detailed analysis of this approach see the work of Altschul and Gish in [3].

2.4 Substitution Matrices

As discussed in section 2.3.2, in order to compare one sequence to another to determine their biological relatedness, we need a scoring system that allows us to compare the likelihood that one amino acid is related to another one. Two primary types of substitution matrices have been created to relate amino acids, Dayhoff's PAM matrices and Henikoff's BLOSUM matrices. The major difference between the two is the use of an explicit evolutionary model versus an implicit one.

2.4.1 PAM Matrices

Dayhoff's substitution matrices rely on the principle of accepted point mutations. An accepted point mutation is the culmination of two processes, the occurrence of a mutation in a gene, and the fixation of that newly-modified gene within a species. The PAM (point accepted mutation) matrix is a measure of the number of accepted point mutations over a period of time; the 1-PAM matrix represents one percent of the amino acids in a sequence mutating and, under conditions when a molecular clock hypothesis holds (see section 4), that amount of change is considered a unit of time [11, 15].

PAM matrices are constructed by examining closely-related sequences, those that have at least 85 percent identity. From those, phylogenetic trees are built using a parsimony method and point changes are recorded from the inferred ancestors of the sequences instead of directly from the sequences themselves (the parsimony method is discussed in depth in section 3). We will refer to these as observed changes, although they are observations of inferred changes, not empirically observed as in the BLOSUM algorithm. This process of tabulating observed changes is illustrated in figure 8.

Dayhoff and colleagues examined sequence data from 34 protein superfamilies grouped into 71 evolutionary trees. They considered a mutation from amino acid X to amino acid Y to have the same probability of Y replacing X. Of the 190 possible amino acid exchanges (20 possible amino acids permuted into groups of 2, divided by 2 since our exchanges are reflexive: 20! / 18! / 2 = 190), 35 never occurred and only 20% of the exchanges involved amino acids whose codons differed by more than one nucleotide.

Once the point mutations were tabulated, Dayhoff calculated the 'relative mutability' to describe the probability that an amino acid will change in a given interval. For a single tree, relative mutability is simply a ratio describing the number of changes for a particular amino acid versus the the total number of times that amino acid has appeared in the tree, or as Dayhoff puts it, been exposed to mutation. If a particular amino acid experiences no changes in a tree, then its relative mutability in that tree is 0. An overall relative mutability ratio is then calculated. The numerator of this ratio is the total number of changes for a single amino acid over all branches of all trees. The denominator is the sum of occurrences over all branches multiplied by the total number of mutations per 100 links for that branch (which normalizes the ratios for various sequence lengths). Next, the frequency of exposure to mutation is calculated for the various amino acids. These values, f_i, describe the background frequency with which each amino acid appears among all the sequences examined and the frequencies of all twenty amino acids sum to 1.

Finally, we have all the necessary information to calculate the 1-PAM matrix, also written as PAM1. The matrix is calculated for each each non-diagonal element, M_{ij}, where M_{ij} represents the probability of amino acid i changing to amino acid j over a unit time:

$$M_{ij} = \frac{\lambda m_i A_{ij}}{\sum_j A_{ij}};$$

m_i is the relative mutability, A_{ij} is the number of accepted point mutations, and λ is a scaling factor. In calculating M_{ij} we multiply the background frequency of amino acid i wby the number of observed mutations of amino acid i to amino acid j, and divide by the sum of all observed mutations of amino acid i. Diagonal elements are calculated by $M_{ii} = 1 - \lambda m_i$ as they represent the probability that no change occurred to the amino acid. In order to calibrate the matrix so that there is one substitution per unit of time we must ensure that the probabilities in each column sum to 1. This is done by choosing an appropriate value for λ to scale the column scores [11].

We can create mutation probability matrices for other distances simply by extrapolating from M by multiplying the matrix by itself. The PAM250 matrix is the PAM1 matrix multiplied by itself 250 times (if λ had been chosen to be larger than a single mutation, then we would have to adjust the distances we would get by multiplying the PAM1 matrix by itself).

Just as we represented scores for single sequences above as log-likelihood ratios, so can we do the same here. As defined in section 2.3.2, a log-likelihood ratio compares the probability of an event under two hypotheses, in this case, as above, we are comparing the likelihood of amino acid i mutating to amino acid j versus amino acid j occurring in a related sequence by chance (the background mutational frequency, f_j). The ratio is expressed as

$$R_{ij} = \frac{M_{ij}}{f_j}.$$

More than a decade after Dayhoff's publication of the PAM matrices Gaston Gonnet recalculated the PAM matrices using the same method as Dayhoff, except with a vastly

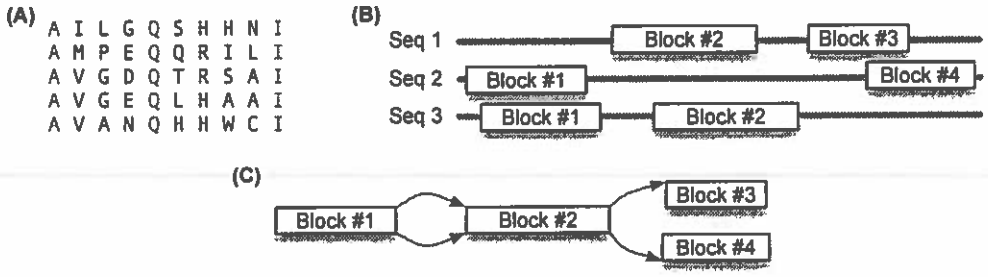


Figure 9: Two stages in the process of creating protein blocks in a family of related proteins. (A) A hypothetical block motif common to five related protein sequences. The motif is defined by a pattern of three amino acids separated by two distances, in this case the pattern is 'A...Q....I'. Based on a figure in [67]. (B, C) An illustration of block assembly. Paths of non-overlapping motif blocks are identified by constructing a graph based on the ordering of blocks among the sequences. Different paths contain different subsets of sequences. (C) Two possible paths through the same set of blocks. The best path (see text) of blocks is found by conducting a depth-first search of the constructed graph.

larger amount of protein sequence data [31]. One of Gonnet's major findings was that Dayhoff's matrices are insufficient for sequences that are not closely related (since the PAM matrices are built from sequences of high similarity). Along with the PAM and BLOSUM matrices, the Gonnet matrices are also commonly used in sequence alignment.

2.4.2 BLOSUM Matrices

While Dayhoff's PAM matrices estimated mutation rates by building phylogenetic trees and counting mutations inferred from the trees, Steven Henikoff and Jorja Henikoff took a different approach with the BLOSUM matrices [33]. While Dayhoff's method did not consider direct changes in actual sequence data, the BLOSUM matrices were constructed by obtaining mutational frequencies directly from relationships represented in protein blocks and does not use an evolutionary model. As we will describe in detail, the BLOSUM approach is to distill a large number of related protein sequences into a series of blocks representing the most commonly occurring subsequences in the set. This involves identifying blocks within individual sequences, combining common blocks between sequences, and then finding a comprehensive set of blocks that provides optimal coverage among all the sequence data. The point mutations and background frequencies used to build the substitution matrix are then determined from this comprehensive set of blocks.

2.4.3 Creating Protein Blocks

When two or more proteins are known to be related the common information between them can be concentrated and utilized to detect more distantly-related members of the protein family. The common information between these proteins can be represented as a series of blocks, short regions of ungapped alignments, separated by unaligned regions [32]. The construction of BLOSUM matrices begins by building a set of protein blocks in the following way.

We start with a group of two or more known, related proteins, such as those available from the PROSITE database and we identify common motifs within the sequences. A motif is a pattern that occurs frequently within a group of related proteins; to discover motifs, the algorithm attempts to identify a particular pattern of amino acids – a set of three amino acids with two intervening distances between them. Such a pattern is illustrated in figure 9 (A). To find these patterns, the algorithm simply scans each sequence once and records all the possible three amino acid patterns. The size and number of motifs that will be detected depends on several parameters, primarily the size of the intervening distances allowed between the pattern's amino acids. Blocks that occur frequently are assigned a block score. This score is calculated first by splitting the block into columns; then, every pair of amino acids in a particular column is scored according to the PAM-250 matrix. Scores for an entire column are averaged together and then the column scores are summed to create a motif block score. Next, block motifs that have been identified are extended in either direction as long as the extensions increase the block score. Overlapping motifs are merged [67].

At this point we have a family of related protein sequences and we have identified all of the motif blocks that occur within those sequences. This result is illustrated in figure 9 (B). To define the final set of blocks that we will use to build our scoring matrix, we need to find the optimum path through our set of blocks that best covers the family of sequences (figure 9 (C)). We do this by constructing an acyclic, directed graph out of the blocks. If block #1 precedes block #2, and does not overlap, then we draw an arc from node #1 to node #2 in our graph. Different paths through the blocks may contain different subsets of sequences; these paths are identified via a depth-first search and a score is calculated for each one. This score is determined by summing, for each block in the path, the number of merged motifs multiplied by the block score, with the total multiplied by the proportion of sequences in the path. The highest scoring path is selected resulting in a final set of protein blocks [32].

2.4.4 Deriving the Frequency Table and Substitution Matrix

For the BLOSUM matrices, Henikoff and Henikoff examined several hundred protein groups creating a database of more than 2000 blocks using the procedure described in the previous section [33]. Once they had obtained the blocks, they then constructed a frequency table. This table is created by counting up all the possible pairs of amino acids for each column in a block.

For example, if the first column of the first block contains 9 'A' residues and 1 'S' residue, then there are 9 'AA' matches and 1 'AS' mismatch. Next, counts of all possible pairs in each column are summed. If a block has a width of w amino acids and a depth of s sequences, then it contributes $ws(s-1)/2$ amino acid pairs to the total count. The result is a frequency table listing the number of times each of the 210 different amino acid pairs occurs among the blocks. The frequency of a pair of amino acids i and j is then denoted as f_{ij}.

Just as was done for single sequence scoring schemes and for the Dayhoff matrices, the table of pair frequencies is converted into a log-likelihood ratio. Recall a log-likelihood ratio compares the probability of a pair of amino acids appearing due to an accepted mutation versus the pair appearing due to their normal background frequency. First, we calculate the

observed probability of occurrence for each amino acid pair, i, j, in the following way:

$$q_{ij} = f_{ij} / \sum_{i=1}^{20} \sum_{j=1}^{i} f_{ij}.$$

Next, we calculate the expected probability of occurrence for each i, j pair. We do this first by calculating the expected probability of the occurrence of amino acid i in an i, j pair:

$$p_i = q_{ii} + \sum_{j \neq i} (q_{ij}/2)$$

and then we combine the expected probability of each component of the pair to get the final probability: $e_{ij} = p_i p_j$, for $i = j$, and $e_{ij} = 2 p_i p_j$ for $i \neq j$. Finally, the score is expressed as a log-likelihood ratio in the following way:

$$\sigma_{ij} = \ln(q_{ij}/e_{ij}).$$

Once all the ratios are calculated, we have the basic form of the BLOSUM substitution matrix.

To reduce multiple contributions to amino acid pair frequencies from the most closely related members of a family, Henikoff and Henikoff clustered the sequences within blocks that share a certain percentage of their amino acids. For example, if sequence segment A is identical to sequence segment B at greater than 80% of their aligned positions, then A and B are clustered and their contributions are averaged when calculating pair frequencies. If sequence segment C is identical to either A or B at greater than 80% of its aligned positions, then A, B, and C are clustered even if C is not 80% similar to both A and B separately. So, for the BLOSUM80 matrix all blocks that are aligned at at least 80% of their positions are clustered, and similarly, the blocks in the BLOSUM62 matrix are clustered when they have at least 62% identity. The clustering process creates a series of matrices for sequences that have diverged by differing amounts. The BLOSUM80 matrix is more useful for sequences that have diverged recently, since clustering sequences preserves conserved regions and discards variation in the composite sequences. Conversely, the BLOSUM62 matrix is more useful for distantly related sequences, since it clusters fewer sequences, preserving the variation in those sequences.

2.5 Sequence Alignment Case Study: BLAST

The Basic Local Alignment Search Tool, or BLAST, is the most widely used sequence search and alignment program in use today. BLAST's central role in bioinformatic analyses is akin to the TCP/IP protocol and its various implementations in the field of network research. Written by Stephen Altschul and colleagues, BLAST was first released in 1990 and later revised in 1997; it continues to enjoy wide use today. BLAST combines a fast, heuristic algorithm to identify potentially homologous subsequences along with each the major concepts we have discussed in this section: alignment through dynamic programming, a scoring system based on substitution matrices, and the determination of relevancy from general scoring statistics. The efficiency of its heuristic approach allows BLAST to compare a query sequence against a database of thousands (and even millions) of sequences identifying segments of sequences in the database that show homology to the query [4]. As opposed to an optimal, global alignment that utilizes a particular scoring system to optimally align

one whole sequence to another, local alignment algorithms align portions of two sequences, or subsequences, that show high-levels of similarity – regions of low conservation do not contribute to the overall measure of similarity.

We defined a general scoring system in section 2.3.2 that allows us to determine, for a particular set of sequences, or a database of sequences, how high an alignment must score for it to be statistically significant. We will call that score S. BLAST's strategy is to minimize the time spent on sequence regions whose score has a low likelihood of exceeding S. It does this by using a *word pair*, which Altschul defines as a short segment pair of fixed length w, that is, a gapless sequence segment present in both the query and the database sequence. BLAST compares each sequence in the database against the query to determine if the sequences contain a word pair with a score of at least T. When it finds such a word pair BLAST then tries to extend the initial segment pair into a larger alignment whose score is at least S. The sensitivity and speed of the algorithm can be largely controlled by changing the parameters T and S.

In more detail, the BLAST algorithm has three phases, compiling a list of high-scoring words, searching the database for occurrences of these words, and extending the word pairs into larger alignments. For protein searches, with a word length of four, there are 20^4 possible words; however, due to the number of possible words there is some discretion in the creation of the word list. A word is only placed on the list if a comparison of it against a word in the query sequence generates a score that surpasses the T threshold. According to Altschul, useful values for the word length w and the score threshold T result in approximately 50 words on the list for every residue in the query sequence. For example, with a query sequence of 250 amino acids, 12,500 words would be generated.

Searching the database for words is a common Computer Science problem of searching a long string for the occurrences of short sstrings. One method to do this would be to create an array with 20^4 elements (assuming $w = 4$). Each word would index into the array and provide a list of all occurrences of that word in the query sequence. As we scanned the database each database word would directly lead to a list of all hits of that word in the query sequence. Once a list of word pairs has been obtained, extending a hit is done in a straight forward way. The algorithm lengthens a hit in either direction until the segment pair score falls significantly below the score of another, already found MSP that has a shorter length. This causes a small reduction in accuracy, but it is a negligible trade-off for the improvement in speed.

For nucleotide sequences, the default word length is $w = 12$. So, given a query sequence of length n, the first word consists of nucleotides in positions 0-11 of the query sequence, the second word consists of positions 1-12, yielding $n - w + 1$ words. There is one additional issue that must be dealt with when using nucleotide sequences with BLAST. DNA sequences are highly-nonrandom, containing biased base compositions and repetitive elements. If a query sequence contains one of these highly repetitive segments then a search will return many insignificant results. To filter these segments from possible search results, BLAST pre-processes its database before any searching is actually done. During creation of the BLAST database the database creation program tabulates the frequencies of all words of length 8. Those that occur much more frequently than by chance are placed on the filter list. Then, when preparing the list of words for a nucleotide query prior to a search the filter list is consulted. If a word from the query sequence appears on the list, it is dropped from the query.

In 1997, Altschul and colleagues released a new version of BLAST, bringing two major refinements to the algorithm [5] to account for the growing gap between sequence database

size and available processing power to search it. Altschul found that lengthening local alignments accounted for more than 90% of BLAST's execution time. To increase speed, the criteria for extending word pairs was changed. While the original BLAST identified short word pairs with an alignment score of at least T, and then attempted to extend those pairs until the alignment score dropped below an already existing shorter score, the new version defines a window of length A. A hit is only extended if a second, non-overlapping word is found within the distance of the window reducing the number of word pairs that will be extended by BLAST. To offset the loss of sensitivity the T parameter was lowered increasing the possible number of words. However, since only a small number of these words are extended, computational time is conserved overall.

The ability to use gapped alignments via dynamic programming in the alignment extension phase was the second major improvement to BLAST. We can imagine a dynamic programming matrix with a diagonal of length w filled in from a word pair that BLAST has identified in the query and database seqeunce. Previous versions of BLAST masked off all but the two, neighboring diagonals of the matrix and limited extending the alignment to this small region. While this conserved computational time, the result of this behavior was that the algorithm could only align small, highly similar regions of the two seqeunces, and the algorithm would often identify several similar regions in a single pair of sequences. It then relied on the use of sum statitics (described in section 2.3.2), to determine the significance of this full set of short alignments. The main problem with using sum statistics was missing any one of the short alignments in the set could cause the combined result not to be identified. By allowing the full use of the dynamic programming matrix, providing for true, gapped alignments, identifying any one of the significant subalignments would allow the recovery of the entire sequence alignment.

In the field of bioinformatics, BLAST is virtually ubiquitous. Utilizing a fast, heuristic search to identify potentially realted sequences, a dynamic programming algorithm to align seqeunces based on scores from a substitution matrix, as well as a robust method to determine statistical significance, BLAST is an effective tool to establish homology between sequences. In the case of genome duplication, we use BLAST extensively to identify paralogous and orthologous gene relationships. As the remainder of this paper will bear out, every other analysis starts by identifying a set of related sequences, and these relationships are determined with BLAST.

2.6 Multiple Sequence Alignment

In the previous section we discussed how to align a pair of sequences and we presented a program that, given a query sequence, could search a database of sequences to find homologous matches. If aligning a pair of seqeunces is useful, it follows that aligning larger numbers of sequences would also be useful. For example, returning to our theme of genome duplication, if I start with a pufferfish sequence, and through two successive BLAST searches identify a paralogous pufferfish seqeunce and an orthologous human sequence, I can not simply put the three sequences together and expect them to line up properly. Obviously, the human gene diverged much further in the past and is likely to have encountered many more substitutions than the parologous pufferfish seqeunce, and natural selection will be placing different constraints on a human gene than on the copy of a pufferfish gene. Once we have identified related genes with a program like BLAST, it is common to then re-align them all together to find a more optimal configuration of the sequences. If we care to do any further analysis, such as building a phylogenetic tree, a multiple alignment is a prerequisite

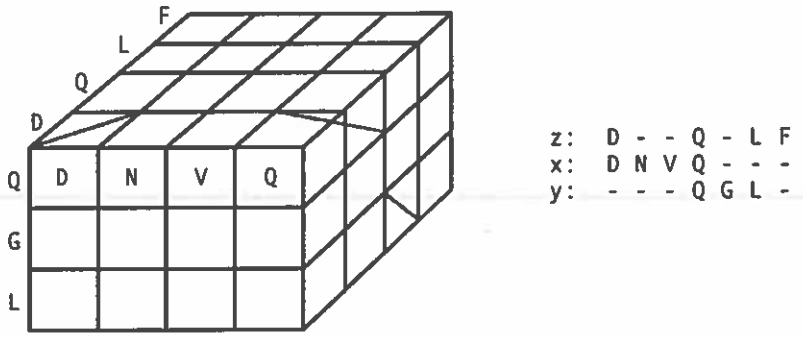


Figure 10: An illustration of a multiple sequence alignment for three sequences, from [7].

– without a proper mulitple alignment it is impossible to infer how the forces of evolution have affected a set of genes.

In section 2.1 above, we described the dynamic programming method to find an optimal alignment between two sequences, given an alphabet and scoring system. To briefly summarize, the algorithm took a pair of sequences of lengths m, and n, and used a two-dimensional matrix to represent all possible alignments of those two sequences. The value of each cell (i, j) of the matrix was calculated using only the data from its nearest neighbors, $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$, representing adding a gap to the first sequence, adding a gap to the second sequence, or extending an existing alignment, respectively. The final alignment was determined by following backpointers from the terminal cell of the matrix, (m, n), back to the first cell reflecting the choices we made when filling in the cells.

The dynamic programming method is conceptually easy to extend to handle additional sequences, and we do so by generalizing our two-dimensional matrix to a k-dimensional one to handle k sequences. The value of each cell is still determined solely by the values of the nearest neighbor cells; however, the number of neighbors increases with the dimension of the matrix. The path indicating the optimal alignment still traverses the full matrix, from the terminal cell back to the initial cell, but can move through the additional dimensions and we may be extending the alignment in several sequences while adding gaps in others. An example of a three dimensional matrix and its associated alignment path is illustrated in figure 10 for three short amino acid sequences.

The major obstacle to using a k-dimensional dynamic programming algorithm is that the computational time required to execute the algorithm rapidly becomes intractable. While optimally aligning two sequences using a dynamic programming method is bounded by $O(nm)$ or, $O(n^2)$ when $n \sim m$, alignment of k sequences is bounded by $O(n^k)$. For that reason, multiple sequence alignment algorithms use heuristics to prune the search space and we will examine several of those algorithms below.

2.6.1 Optimizing Multiple Sequence Alignment through Heuristics

In [7], Humberto Carrillo and David Lipman described the multiple alignment problem and proposed a method to optimize the dynamic programming algorithm. Their optimization hinges on the idea that the multiple alignment of k sequences implies $\binom{k}{2}$ pairwise alignments of those sequences. Any scoring scheme to align them all together must be

monotonically increasing and can not be less than any single pairwise comparison. Therefore, Carrillo and Lipman assert that a multiple alignment will not move beyond the space in the k-dimensional matrix that is defined by the individual pairwise alignments. So, the computational requirements of a multiple sequence alignment would be a function of the size of the subregion of the k-dimensional matrix that defines the space used for the individual pairwise comparisons plus the number of computations necessary to generate it. This idea was implemented by Lipman and colleagues in [53] and they found that the computational demands of dynamic programming were greatly reduced, allowing them to align as many as eight sequences simultaneously (compared against a limit of three sequences for a standard dynamic programming algorithm when the paper was published in 1989).

2.6.2 Progressive Sequence Alignment

In [25], Da-Fei Feng and Russell Doolittle propose the first workable heuristic method to create multiple alignments, known as progressive sequence alignment. Their work was completed as a means to create more accurate phylogenetic trees. Previous schemes to construct trees relied on creating a topology by classifying a set of sequences according to their differences using the principle of parsimony. The best trees were thought to be those that could account for the differences in sequences by the smallest number of genetic events. Tree creation was done by aligning all the sequences pairwise and then clustering the sequences according to their similarities, often by hand. Grouping the pairwise-aligned sequences was problematic; when sequence A was compared with sequence B gaps would appear, but when A or B was aligned with C, the arrangement of gaps would be entirely different. People dealt with this problem by 'eyeballing' the sequences and shifting them to make them fit by hand.

Feng's method, on the other hand, follows the rule "once a gap, always a gap" and aligns sequences beginning with the most closely related pair (according to a pairwise alignment score) and proceeding one at a time from there. The progressive method works in the following way. First, make all $n(n-1)/2$ pairwise alignments using a standard dynamic programming method, scoring the alignments with the PAM substitution matrix and recording the results. Since this work was published before Karlin and Altschul's general scoring statistics were widely used (see section 2.3.2), Feng instead compared each of the pairwise alignment scores against a set of randomized sequences to determine their statistical significance. The 'difference scores' were then used to provide an order for proceeding with the progressive alignment. The algorithm then adds one sequence at a time to the multiple alignment, allowing new gaps to be incorporated with each additional sequence but preserving pre-existing gaps. Given two components of the multiple alignment, A and B, C is added to the alignment in a way that maximizes the total score, trying $(AB)C$ first, and then $C(AB)$, keeping the alignment that results in the best score. This process is repeated for each added sequence or sequences. The final alignment is scored by taking the completed multiple alignment and randomizing it while preserving the gap locations. The final alignment is then compared against the randomized multiple alignment.

2.6.3 Case study: CLUSTALW

Just as BLAST is the *de facto* tool for finding homologous sequences, CLUSTALW fills that space for multiple alignments. Released in 1994, Julie Thompson and colleagues describe the program in [72]. Thompson outlines a series of improvements to the progressive sequence

alignment algorithm originally published by Feng and Doolittle in [25] after highlighting two major problems with progressive sequence alignment.

One of the major problems that can happen when using a progressive alignment algorithm, or as it would be known in Computer Science, a *greedy* algorithm, is that mistakes made early in the process can constrain the possible outcomes of the final alignment. This may result in the algorithm finding a local minimum instead of finding the true, optimal alignment. These mistakes usually arise from adding sequences to the alignment in the wrong order. If you add a more distantly related sequence before a more closely related seqeunce, than you will end up with gaps in the wrong positions, and since gap placement is fixed in the algorithm, the earlier you make the error, the larger the magnitude of the error. A second source of local minima may arise from choices in substitution matrices and gap penalties. In a multiple alignment it is common to have homologous sequences from a variety of organisms, with varying amounts of evolutionary time separating those organisms. We have seen that substitution matrices are tuned to work on sequences at a particular evolutionary distance. So, if you choose a single substitution matrix, it will likely only work well for a subset of the sequences in the multiple alignment. Again, the more distant your sequences, the more crucial it is to choose a proper substitution matrix. Additionally, since gaps do not occur randomly within sequence alignments (they are much more common in between major secondary structures like α-helices and β-strands, for example) the gap-opening penalty and the gap-extension penalty should be adjusted in a position and residue specific manner.

The CLUSTALW algorithm, which accounts for these problems, works in the following way. First, like in Feng's work, sequences are initially aligned in a pairwise manner using either a fast, approximate method or using a full mathematically optimal alignment. Next, an initial guide tree is created using the neighbor-joining algorithm to build an unrooted tree. Although the neighbor-joining algorithm is described in detail in section 3, briefly, the it creates a tree by repeatedly grouping the most closely related sequences based on a measure of distance. In this case, CLUSTALW uses the percent identity (the number of identities in the best alignment divided by the number of residues compared, excluding gaps) from each pairwise alignment as a measure of distance. Once the unrooted guide tree is complete, CLUSTALW then roots the tree by placing a mid-point where the means of the branch lengths are equal on either side of the root. The sequences are then weighted by taking the distance from the root of the tree; since we are using percent identity as our distance, weights for individual sequences vary inversely with their percent identity. In cases where branches are shared, the sequences on the branch equally share the weight.

In more detail, sequence weights are calculated from the guide tree and are normalized so that the largest weight equals 1. Groups of closely related sequences receive low weights because they duplicate information, while more diverged sequences have higher weights. The weights are a measure of the informativeness of a particular site in the multiple alignment. For example, if two very closely related sequences have an amino acid position in common, that event is much less informative than if two very divergent sequences have an amino acid position in common.

Now that we have generated the guide tree and determined the sequence weights, a progressive alignment is performed by aligning sequences in the order of the guide tree. Gaps from earlier alignments remain fixed. In order to calculate the score between a position in the multiple sequence alignment with the corresponding position of the sequence (or a group of sequences) we are adding to the alignment, the average of all the pairwise matrix scores from the amino acids in the two sets of sequences is used. So, if we are adding a group of

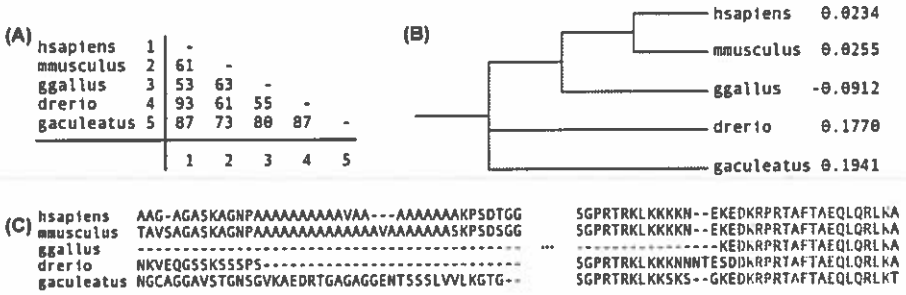


Figure 11: The major steps of the CLUSTALW alignment program, based on an image from [72] with original data; aligning the human engrailed gene (hsapiens), along with the orthologous mouse (mmusculus), chicken (ggallus), zebrafish (drerio), and stickleback (gaculeatus) genes. (A) The scores of the initial optimal pairwise alignments which are the percent identity from the alignment converted into a distance. (B) The generated, rooted neighbor-joining guide tree, with sequence weights displayed at the tips of the branches. (C) Two segments of the progressively aligned sequences with total sequence identity is shown in red.

two sequences to a multiple alignment that already contains four sequences, then we will average the eight pairwise values from the substitution matrix. The weights we calculated in the previous step are used directly in this calculation such that each of the eight values that are averaged together would first be multiplied by the sequence weights corresponding to the two sequences being compared in the substitution matrix.

Initial gap penalties for the final progressive alignment are calculated depending on the substitution matrix, the similarity of the sequences and the length of the sequences. Gap opening penalties (GOP) are then adjusted in a position-specific way. If the alignment is working in an area with a pre-existing gap, the GOP is lowered to encourage gaps in places where there is already a gap. If the current gap is within eight residues of the previous gap, the GOP is increased since gaps do not usually occur too close together. Finally, if the current sequence contains a hydrophilic segment, the GOP is again lowered. Likewise, CLUSTALW switches between 4 different substitution matrices (four variants from either the PAM or BLOSUM series of matrices) depending on the distance of the sequences according to the guide tree.

2.6.4 Other Multiple Alignment Strategies

Multiple sequence alignment is currently a very active area of research, particularly in the area of progressive sequence alignment. As we have seen with Feng and Doolittle's progressive alignment strategy and with the strategy implemented by CLUSTALW, there are three basic components to the progressive sequence alignment algorithm: the generation of optimal pairwise alignments, which are used to construct a guide tree, which is then used to create the multiple alignment. In this section we will discuss some of the more recent strategies for progressive alignment and describe how they differ from the classic approach.

New progressive alignment algorithms seek to improve on the approach by CLUSTALW in different ways. Some algorithms focus on increasing the speed of parts of the algorithm, others focus on improving gap placement, while still others focus on refining the final progressive alignment through iteration. The program MUSCLE, by Robert Edgar, makes many

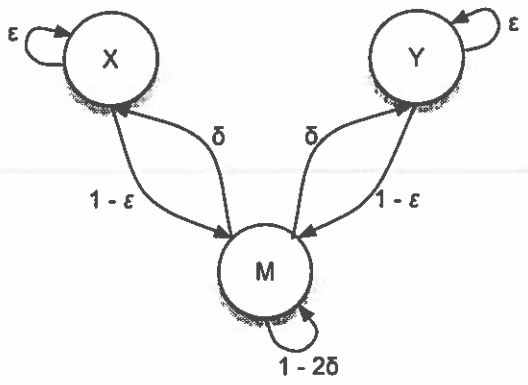


Figure 12: The pair-hidden Markov Model with three states. State M emits two letters and represents an extension of an alignment. States X and Y each emit a single letter and a gap representing adding a gap in one of the two sequences. The gap opening and extension penalties are derived from the transition probabilities, σ and ϵ.

changes to the standard algorithm in hopes of gaining improvements [19]. For example, to speed up the initial pairwise alignments MUSCLE uses k-mer counting with compressed alphabets [18]. Creating a compressed alphabet involves partitioning the standard twenty-letter amino acid alphabet into N disjoint subsets with the goal that substitutions in an alignment are more likely to fall into one subset rather than into two subsets, increasing identity. A k-mer is a letter sequence of length k and is also called a word; related sequences generally have more k-mers in common than expected by chance. Recall from section 2.5 that BLAST counted the number of k-mers common to two sequences as a method to find possible high-scoring sequence pairs. As a measure of evolutionary distance, MUSCLE computes the fractional common k-mer count F, by summing the number of times each k-mer occurs in the pairs of sequences divided by the minimum of the sequence lengths. These distance measures are then used to build an initial tree to guide the progressive alignment. Edgar claims that this measure is more accurate than CLUSTALW's use of percent identity in its initial pairwise alignment stage and that by counting k-mers in a compressed alphabet, instead of performing any actual pairwise alignments, the MUSCLE algorithm gains efficiency in the initial stage of the progressive alignment strategy.

An alternative to increasing the speed of the initial pairwise alignment phase is to try to improve its accuracy. Both Löytynoja and Do, in [55] and [13] respectively, have refashioned their pairwise alignment algorithms into hidden Markov models (HMM). While we will discuss Markov models in more depth in section 3, briefly, a hidden Markov model allows an algorithm to be described as a set of states, with arrows connecting the states together as shown in figure 12. States have transition probabilities that describe the likelihood of moving from one state to another as well as emission probabilities. When we move from one state into another, we emit a residue from that state's emission probability distribution [16] (an in-depth discussion of HMMs can be found in [15]). Both Löytynoja and Do use a three state HMM to perform alignment, where state M represents an extension to an alignment and emits two letters, one from each sequence. States X and Y represent adding gaps to an alignment and each one emits a letter from one sequence along with a gap.

The transition probabilities represent the likelihood of opening a gap, if moving from state M to state X or Y, extending the alignment if moving from M to M, or extending a gap if moving from X to X or Y to Y. Each possible alignment can be thought of as a path moving through a series of states. We want to choose the state path that gives the highest total probability (obtained by combining all the individual transition and emission probabilities occurring along the path). An HMM is a full probabilistic model and it can be manipulated by Bayesian probability theory, allowing parameters to be trained or optimized and providing for measures of score significance (many of these concepts will be explored in more detail in the context of tree building in section 3). While Do simply used the values from the BLOSUM62 substitution matrix to govern emission probabilities, Löytynoja used a full phylogenetic substitution model (such as Jukes and Cantor or Hasegawa) to increase accuracy and power of the pairwise alignments.

A final modification that has been investigated is to add a refinement step after the progressive alignment stage. In both Edgar's MUSCLE and Do's ProbCons after the final progressive alignment is finished, the tree representing the multiple alignment is partitioned and the two subtrees are separately re-aligned. In the case of MUSCLE, the tree is partitioned along each edge; if a particular re-alignment produces a better score, the subtree is kept, otherwise the alignment is discarded. The process stops when no improvements are detected after re-alignment.

2.7 Conclusions

The area of sequence alignment is still very active, having made two towering achievments, and with work continuing in two other major areas. The mathematically optimal seqeunce alignment algorithm, along with its solid statistical foundation, as represented in the program BLAST, has changed the way research is done in the field of biology. Representing a nexus of Computer Science and biological research, BLAST has remained an essential tool since its inception in the early 1990's. Although work continues in new alignment methods, such as Edgar's work on compressed alphabets, BLAST remains dominant in the field. In contrast, multiple alignment algorithms continue to proliferate. Although CLUSTALW is the most widely used multiple alignment algorithm, its reliance on a multitude of different heuristics leaves open many areas for improvement. Unfortunately, many new multiple alignment algorithms, such as MUSCLE, themselves rely on an array of unproven heuristic methods. Similary, while substitution matrices are used continuously, they are still a major weak point in the alignment field. A detailed look at how these matrices are assembled reveals yet another set of heuristics, with both the PAM and BLOSUM matrices built on a set of implied, mixed models of evolution. As a measure of changing the field, the work of those like Löytynoja, who are working to anchor sequence alignment in formal, mathematical models, will be of extreme importance in moving the field in the future.

3 Phylogenetics

While the previous section of this paper focused on identifying genes that shared an evolutionary relationship, this section will examine the area of phylogenetics. Phylogenetics is a set of methods that allows us to infer the ancestral changes in a group of genes known to be related – when novel features appeared, how often they appeared, and perhaps when they were lost – by examining the present state of a gene in multiple organisms and using

evolutionary models along with powerful statistics to work backwards and infer the ancestral states of a particular gene or gene family. For example, the *Pax6* gene encodes a transcription factor that plays an important role in the development of eyes in both humans and in fruit flies [47]. We may want to answer the question: when did the *Pax6* gene appear in evolutionary history? And, how has it changed in different organisms? To do so, we could collect *Pax6* transcripts from a variety of species, some with complex forms of the eye and some with simple forms, and we would construct a phylogenetic tree from those genes. In doing so, we would then be able infer how the gene changed over time and in what lineages those changes occurred. In terms of genome duplication, we would primarily be interested in confirming relationships between genes that we identified through alignments. To do so, we would take the sets of genes we identified as paralogs and orthologs and we would build trees from them looking for a particular topology in the tree that indicates that there was indeed a duplication (we will examine the genome duplication topology in section 5).

At its core, phylogenetic analysis is about building trees; the leaves of the tree representing contemporaneous organisms, or characters of those organisms such as genes or proteins, and a series of branches moving backwards in time to the root of the tree; internal nodes in the tree represent ancestral organisms. Examining the tree from its root out to the leaves describes a precise ordering of speciation, from the ancient ancestral organism, to its modern-day descendants. Organisms, and their corresponding characters such as proteins, evolved by a certain path; the existence of modern organisms are proof that such a path exists. Phylogenetic methods are a means to discover that path, however, when we build a tree the result is a hypothesis for what really happened. Although phylogenetic analysis has been in use for almost 100 years ([51]), the development of rigorous methods has progressed along with the availability of molecular sequence data, and for that reason, we will focus on molecular phylogenetics in this study.

There are four major families of methods for inferring phylogenies that we will examine in detail: parsimony, distance methods, maximum likelihood, and Bayesian inference. These methods differ in several major ways; parsimony is based on an implicit model of evolution, while maximum likelihood (ML) and Bayesian inference methods use explicit models; distance methods greedily construct a tree based on a mathematical distance measure of the data, while parsimony, ML, and Bayesian methods look for an optimal tree among all possible trees; parsimony and distance methods are fast and can process large numbers amounts of data to create large trees, while ML and Bayesian methods are computationally intense and the amount of data they can analyze is limited by the computing resources available. Finally, while maximum likelihood uses a point-estimation of its model parameters, Bayesian methods integrate over all parameter values. We will explore these differences in more detail in this section.

Although the concept of phylogenetic reconstruction has been around for a long time, in the last fifteen years researchers have begun to find consensus around the idea that phylogenetic reconstruction is a problem of statistical inference [35]. For this reason, it is important to examine phylogenetic methods for their statistical properties and assumptions. Good methods should include the following properties. They should be accurate, that is, they should indicate how well a method estimates the correct tree. They should provide some means of falsifying the assumptions made during the analysis (including things such as the use of a bifurcating tree, and the model that describes the evolutionary process). They should provide a means to choose between different evolutionary models. For example, in choosing between simple and complex models for character change, how can one justify the complex model? Finally, they should be able to estimate a level of confidence in the

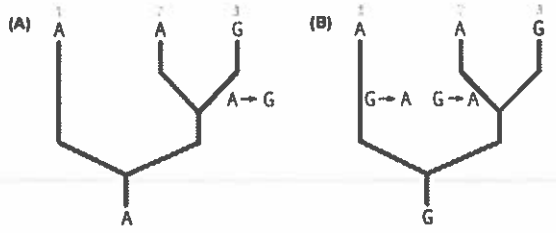


Figure 13: Two possible trees to explain the same nucleotide site in a common gene of three organisms. The first tree (A) requires one substitutions to explain the data while the second tree (B) requires two substitutions. The principle of parsimony would select tree (A) as the best tree, since it requires fewer changes to describe the data.

phylogeny. In the remainder of this section we will describe the major phylogenetic methods and try to assess them in terms of the properties listed above.

3.1 Parsimony Methods

If we look at a particular site in a gene shared by three organisms, and in the first and second organism the site contains the nucleotide A, but in the third organism the same site contains the nucleotide G, we may wish to ask the question of which nucleotide represents the ancestral state? One possible answer is that the last common ancestor of the three organisms had an A at that site, and in the time since speciation a G replaced the A in the third organism. Another possibility is that the ancestral organism actually had a G at that site and both the first and second organisms experienced a substitution. In fact, there are an infinite number of possible substitutions that could lead to the present orientation of the nucleotides at that site.

More broadly, we can comparatively use our sequences to construct a tree relating the organisms. Each leaf on the tree would contain the nucleotide (or amino acid or character) at that site for one organism. The internal nodes of the tree would represent the ancestral state of that site. The scenario described above could then be represented as two different trees, with each tree containing a different nucleotide at its internal ancestral node. The question then becomes, which tree provides the best explanation of the data? This scenario is illustrated in figure 13.

The principle idea of parsimony states that the tree requiring the minimum number of changes would be selected as the most parsimonious tree. In our example, the first tree (figure 13 (A)) would have a single $A \rightarrow G$ change while the second tree (figure 13 (B)) would contain two $G \rightarrow A$ changes – parsimony dictates that we choose the first tree. As Joseph Felsenstein puts it, the number of extra state changes on a tree counts the number of ancillary hypotheses that must be erected to explain evolution in the group [24].

As a brief aside, to reconstruct the tree, we are interested in informative sites in the nucleotide sequences [51]. In general, informative sites give different support for different trees, while uninformative sites give equal support for all trees. The comparison of informative sites will allow us to differentiate between the many trees that may explain our data. In the case of parsimony, for example, if at the first site of a set of sequences all taxa share the same nucleotide, then that site provides us with no information as to the evolutionary relationships between our organisms. We may choose to use only these informative sites to

build the tree in order to save computational resources.

Given a set of related nucleotide sequences and a tree topology presumed to describe their ancestral relationships, Walter Fitch's method, published in 1971 in [26], can determine the exact number of nucleotide replacements that are the minimum necessary to account for the descent of the sequences from their last common ancestor. In other words, given a particular tree topology, Fitch's method provides a score of optimality for that particular tree. However, to find the most optimal tree, we must search the space of all possible trees and that space can be very large. Due to tree space size we must rely on heuristics; after describing Fitch's algorithm in detail, we will look at two algorithms for searching the tree space.

Fitch's parsimony algorithm begins with a set of related nucleotide sequences, as well as a known topology of the tree relating the organisms. The preliminary phase is concerned with reconstructing the internal nodes of the tree and it assigns values to the internal nodes by working from the leaves of the tree down to the root (a postorder traversal). The tree is initialized to have a length of 0. The contents of an internal node are decided on the following criteria: for any particular internal node in the tree if there are common nucleotides among its descendants, then the value of that internal node is the intersection of those nucleotides. If there are no common nucleotides, then we assign the union of the descendant nucleotides to the internal node and we increase the length of the tree by one. This process is illustrated in figure 14 (A). It should be noted that for every union in the preliminary tree, a mutation must have been fixed at this nucleotide position in one of the descendants. Counting the number of unions will give the minimum number of substitutions required to account for all changes from the last common ancestor in the tree.

While the preliminary phase moved from the leaves of the tree to the root, the final phase moves from the root to the leaves (a preorder traversal) and is concerned with correcting for several situations in which not enough information was available in the previous phase. The final phase of the algorithm is governed by three rules:

- **Rule of diminished ambiguity.** Remove nucleotides from an internal node if its immediate ancestor does not require them. An intersection operation higher in the tree may have made certain nucleotides at the current internal node obsolete. An example of this process is illustrated in figure 14 (B).

- **Rule of expanded ambiguity.** If the internal node was formed by a union operation add nucleotides that are present in the immediate ancestor but missing in the current node. A union operation higher in the tree may have added more ambiguity as to when a particular character originated in the tree. An example of this process is illustrated in figure 14 (C).

- **Rule of encompassing ambiguity.** If an internal node was formed by an intersection add nucleotides to the node if that node's ancestor has them and at least one of its descendants has them. A union operation higher in the tree may have made the intersection operation at the current node too strict. An example of this process is illustrated in figure 14 (D).

Once the final phase is completed, we are left with all the possible nucleotides that may occupy each internal node of a single tree topology. Since there may be more than one nucleotide at each internal node in our final reconstruction, we are left with one or more equally parsimonious trees; any additional nucleotides would require the postulation of

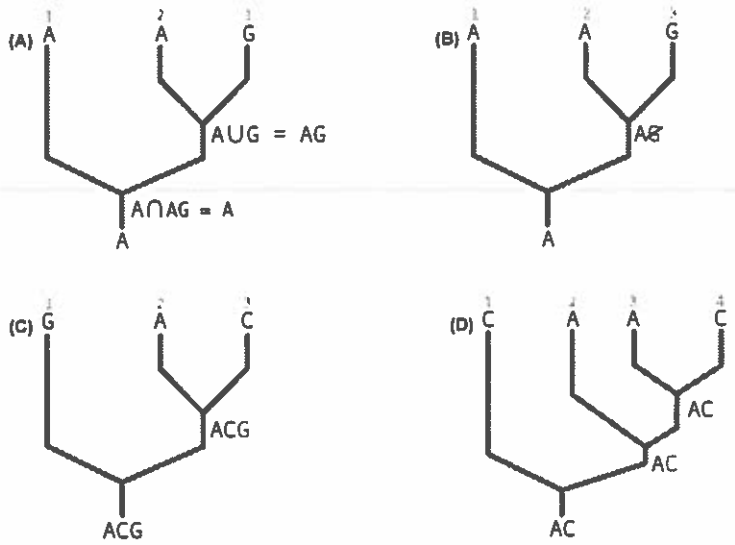


Figure 14: Fitch's parsimony algorithm. (A) The preliminary stage processing from the leaves of the tree to the root. If two nodes have common characters, take the intersection, otherwise take the union. (B - D) The final phase of the algorithm. (B) Prune out characters from internal nodes if the ancestral node does not contain them (red strike-through). (C) If an internal node was formed by a union, add any characters present in the ancestral node (green). (D) If an internal node was formed by an intersection, add any characters present in the ancestor if it is also present in at least one descendant (green). Based on Fitch's algorithm as described in [26].

more substitution events violating our parsimony principle. However, one last step involves enumerating the possible linkages within this set of equal trees. When enumerating the particular nucleotides in our tree, a nucleotide in an ancestral node must link to the same nucleotide in a descendant node if it is available, otherwise, the ancestral node may be linked to any of the nucleotides in the descendant node.

3.1.1 Tree Enumeration

Fitch's algorithm is still widely used as a method to find an optimality score for a particular tree topology, however, there are many possible tree topologies that must be investigated. The simplest algorithm for enumerating trees is to simply perform an exhaustive search, but, the number of trees to search quickly makes this approach intractable. For example, we start with an unrooted, minimal three-organism tree. To add a forth organism to the tree there are three possible branches that it could be added to for a total of three trees. Adding a fifth organism generates fifteen possible trees and in general adding the ith organism to the set of trees for $i-1$ organisms yields $2i-5$ branches on which the new organism could be placed. The first two sets of trees are illustrated in figure 15. Ten organisms gives over two million trees and twenty organisms gives over $2x10^{20}$ possible trees [69]! We will briefly look at two alternative algorithms to generate trees, the heuristic bisection-reconnection searching method as well as a branch-and-bound method.

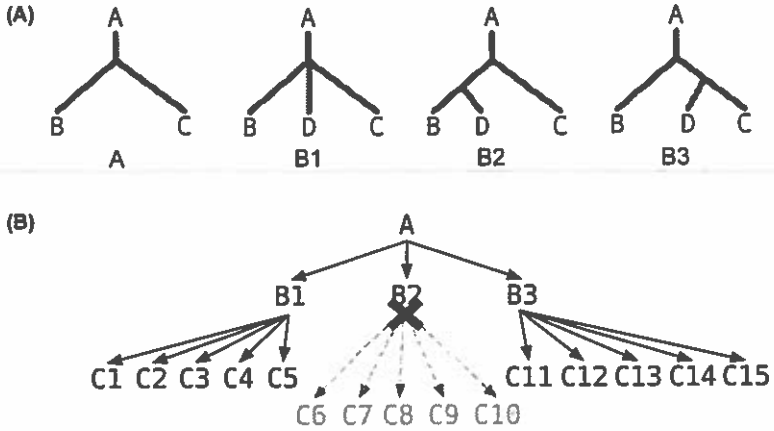


Figure 15: Enumerating trees and the branch-and-bound algorithm. (A) The minimum, three organism, non-rooted tree along with all possible ways to add a fourth organism to tree A. (B) An illustration of the branch-and-bound algorithm. Trees are searched in a depth first manner. If at any time a tree's score exceeds the bound L, then the algorithm stops descending and moves on to the next node. In this case, tree $B2$ exceeded the bound, so trees $C6$ to $C10$ will not be checked; instead the algorithm will move back up to node A and then onto node $B3$. Based on an image in [69].

Bisection-reconnection searching, as discussed by Ronquist in [63], takes an initial tree and clips it into two or more components. The subtrees are then reconnected at all possible positions and the length of each rearrangement is compared to that of the original tree using Fitch's optimality algorithm. If a tree of the same length as the original is found, the new tree is added to the tree set in memory, however, if a shorter tree is found, then the trees in memory are deleted and a new round of swapping is initiated on the shorter tree. The search halts when all rearrangements have been tried on all trees in memory and no additional trees of the same length or shorter can be found.

In more detail, the algorithm proceeds in the following way. First, an initial tree, which is a reasonable approximation of the true tree, is determined. This can be done through any method, such as one of the distance methods described below. The initial tree is then scored via Fitch's algorithm, assigning internal branch nodes and calculating branch lengths. Each node in the tree has a state set, or, the set of nucleotides that could occupy that node in the tree. The initial state set corresponds to the nucleotides present after phase one of Fitch's algorithm, while the final state set corresponds to the nucleotides present after the second phase of Fitch's algorithm.

Next, the tree is clipped, or split into a source tree and a target tree (the target tree contains the former root node) and a potential root node is chosen in the source tree. The two subtrees are once again scored and then, the target tree is reattached to the source tree at every possible branch. Each possible root node in the source tree is attempted, and with each root node, every branch in the target tree is reconnected. Finally, when all potential root nodes have been tried in the source, and all perturbations with the target tree have been tried, the algorithm returns to the original tree and starts again with a new clipping. At any point, if a shorter tree is found, the tree is recorded and the process starts again.

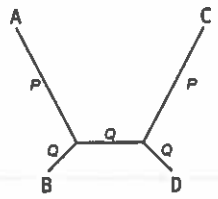


Figure 16: An example for which parsimony methods fail to remain consistent, commonly known as long-branch attraction, P and Q represent different rates of evolution, $A - D$ represent a character or nucleotide site in four species. Based on an image in [21].

A second method of tree enumeration is the branch-and-bound algorithm as described by Swofford in [69]. The branch-and-bound algorithm works in exactly the same was as an exhaustive search with one major exception: by bounding the search major subsets of trees that exceed the bound can be skipped saving execution time. The algorithm starts with the minimum three organism tree as well as an upper bound score L which can be generated initially from a random tree. It then proceeds adding organisms to the tree in a depth first manner. So, the initial three-organism tree A will generate three four-organism trees, $B1$, $B2$, and $B3$. Adding a fifth organism to tree $B1$ will generate five more trees $C1$ through $C5$ and so on. This process is illustrated in figure 15 (B). If at any time the score of a tree exceeds L, the algorithm stops descending along that path and backtracks up a node and starts down the next path. This is because if a tree's score is larger than L adding additional organisms to the tree cannot decrease the score. If at anytime the algorithm adds all organisms to the tree and finds a new bound smaller than L, then it replaces L as the bound and the algorithm continues searching. It is easy to see that in the worst case scenario the algorithm will be the same as an exhaustive search but it can generally handle twenty or more organisms.

3.1.2 Parsimony Inconsistencies

A statistical estimation method is consistent if it approaches the true value of the quantity as larger and larger amounts of data are accumulated. For example, the mean of normally distributed set of data gets closer and closer to the true population mean as the amount of data increases [24].

Parsimony methods do not posses the property of consistency in all cases. One well known counter-example, a problem commonly labeled long-branch attraction, involves a four-species tree with unequal rates of evolution among two groups of lineages. The tree is composed of four species, A, B, C, and D and the topology of the tree looks like the following: ((A,B)(C,D)). In this tree, species A and D have long branch lengths, while the others are short. As we collect more sites, the method will choose the erroneous tree (A,D)(B,C) with probability approaching 100% [22, 24]. This tree is pictured in figure 16.

To prove the point emphatically, Felsenstein published a paper in 1978 detailing how this inconsistency happens [21]. In the paper, he constructs a particular three-species case in which lack of consistency can be proven.

The example is formulated in the following way. The tree is composed of three species, A, B, and C. In the tree characters can have two states, 1 or 0 (as opposed to the conventional four states in a nucleotide example) and a character site can change from 0 to 1, but it

cannot be reversed. The probability that a character that was in state '0' and changes to '1' is represented by three rates, P, Q, and R. The two short branches in the tree share rate 'Q', while the two long branches share 'P'. The root branch has rate 'R'. These rates are meant to model common differences in the rate of evolution among species.

The final state of the leaves in the tree is then represented as a 3-digit binary number. '000' represents no changes in the tree, while '011' represents species A as having experienced no change, and with B and C both having changed to a 1. Given the rates of change on the branches, we can then calculate the probability of each set of possible character states, written as $P_{000}, P_{001}, P_{010}, ...P_{111}$. So, if you examine N different characters in the three species you can count how many of them produce each of the eight possible trees. n_{010} then represents the number of characters that give a '010' tree configuration. So, the parsimony method will estimate the correct phylogeny as (AB)C, for example, as long as $n_{110} >= n_{101}, n_{011}$. According to the law of large numbers, as the number of characters examined, N, approaches infinity, then $n_{ijk}/N = P_{ijk}$, or, the number of character states predicting a particular tree will approach the probability of that tree.

The method will be consistent as long as $P_{110} > P_{101}, P_{011}$ and, in general, whichever of these probabilities is greatest will determine which of the final trees will be selected as the proper phylogeny. Likewise, if the probabilities do not hold, and $P_{101} > P_{110}, P_{011}$, then as we look at more characters the method will give the wrong phylogeny with increasing certainty. By using the known formulas for the probabilities of particular trees, Felsenstein demonstrates, that when the rates P and Q are sufficiently different, or when the difference in length of the long and short branches is too great, the equality does not hold and the method fails to produce the correct tree.

3.2 Distance Methods

A second class of methods used to construct phylogenetic trees is known as the distance methods. Distance methods use a clustering algorithm to group pairs of species together based on some measure of distance. Generally, a matrix is constructed from pairwise distances between species, for example, when using nucleotide data, the distance may be the fraction of sites different between the sequences of two species. A phylogenetic tree is then built stepwise by successively grouping the most closely related species in the matrix. This tree predicts the distance between each set of species as the sum of branch lengths between the species in the tree [24, 51]. One important distinction that separates the distance methods from parsimony, maximum likelihood, or Bayesian inference is that the distance methods present an algorithm to build a tree, while the other three methods provide a method to determine the optimality of an already known tree.

A statistical measure of goodness of fit, often a least squares measure, is applied to compare the observed distances with the expected distances in the tree. The phylogeny that minimizes this difference is the preferred tree. A simple example of this method is the unweighted pair-group method with arithmetic mean (UPGMA). This algorithm works almost exactly as described above, constructing a matrix of pairwise distances between all of the species from which to build a tree. If for example, we have four species, A, B, C, and D, and A and C have the smallest measure of distance between them, then they are combined into a composite species and a new matrix is created, (AC), B, and D. A and C are placed at the leaves of a bifurcating branch in the tree and the branch length becomes the average distance between them: $d_{AB} = (d_a + d_b)/2$ [51]. This algorithm is instructive, but not very accurate [64], and is no longer in serious use.

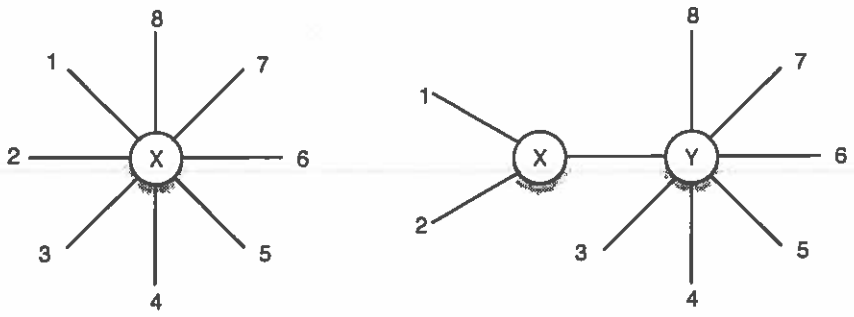


Figure 17: An example of the neighbor-joining method. On the left is the initial, unrooted star tree. On the right is the tree after the algorithm completes its first round of clustering. Organisms 1 and 2 were chosen as neighbors since they had the smallest least-squares estimate of branch lengths among all pairs in the tree. Based on a figure from [64].

A second example that is still used today, although generally as a method to find a reasonable preliminary tree topology, is Saitou and Nei's neighbor-joining method. In a general data-mining context this algorithm is often referred to as hierarchical clustering. Saitou and Nei applied this algorithm in a biological context as a response to the inefficient parsimony methods that preceded it. As we saw in section 3.1, parsimony methods examine large numbers of tree topologies to uncover the tree or trees with the minimum evolutionary change. The neighbor-joining algorithm is not guaranteed to find the tree of minimum evolution, but it will identify the topology and the branch lengths of a unique tree efficiently.

The neighbor-joining method begins with an unrooted tree in a star topology. The minimum distance between species in the tree is determined by computing a least-squares estimate of branch lengths for each pair of species in the tree. Similar to fitting a curve to a set of empirical data, the least-squares estimate minimizes the sum of deviations between branch lengths. The pair of species that minimizes this value are chosen as neighbors and this pair of species is then regarded as a composite unit in the tree. The algorithm continues in a greedy fashion, continuing to group pairs of species until the number of composite species becomes three, and there is only one unrooted tree. The initial star tree and the first clustering step in this process are illustrated in figure 17 [64].

3.2.1 Distance Method Inconsistencies

As mentioned earlier, a method of estimation is consistent if, as more data accumulates, the estimated value converges to its true value. Distance methods are generally consistent when using sequence data, however, there is a problem that stems from the fact that evolutionary distance is represented as the sum of branch lengths between nodes in the tree. For example, if we expect changes in 10% of the nucleotides between nodes A and B in a tree, and also 10% of the nucleotides between nodes B and C, then we expect that 1% of the sites have changed twice between A and C. Under a model where all nucleotides have an equally likely chance of a substitution, one-third of the changes will be a reversion back to the original nucleotide. So, the net difference between A and C is not 20%, but rather 19.67%. As two DNA sequences become very far apart in the tree, the sum of branch lengths between

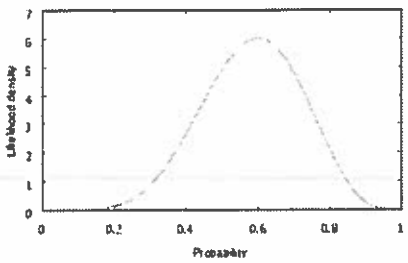


Figure 18: An example of the likelihood surface as generated by a coin-tossing experiment. The likelihood is maximized when the probability is equal to the proportion of heads that appeared during the experiment. The figure is based on [35].

then should rise to infinity, but it cannot rise above 100%. Taking back-substitutions into account, the lengths will approach 75% [24]. Thus, when parallel or back-substitutions are prevalent, problems of inconsistency can cause branches to be too short in the estimated trees.

3.3 Maximum Likelihood Methods

Maximum likelihood is a general method of deriving statistical estimates of parameters. As with parsimony, maximum likelihood can be used to determine the optimality of a given phylogenetic tree. The method calls for an evolutionary model and a set of data that can be sequences of nucleotides, codons, or amino acids. The model, given a number of parameters, specifies how substitutions occur along the lineages in the tree and the data is the empirically collected sequence data for a set of organisms. The likelihood of a tree is the probability of the data given the tree and the model and can be expressed as $P(D|T, M)$. The probability of observing the data under the assumed model will change depending on the particular parameter values for the model. Maximum likelihood chooses the combination of parameter values over all parameters that maximizes the probability of observing the data [22, 24, 35].

As a simple example of how the maximum likelihood method works, consider a coin tossing experiment. Assume the probability of observing heads while tossing a coin is unknown and we would like to estimate it (perhaps to check its fairness). The method requires the observed data and a model describing the probability of observing the data. A good model that describes the probability of observing h heads out of n coin tosses is the binomial distribution, which describes the number of successes in a series of independent success or failure experiments. The binomial distribution is defined as

$$Pr[h|p, n] = \binom{n}{h} p^h (1-p)^{n-h},$$

where p is the probability of observing heads. The likelihood in this case, then, is p, the probability of observing heads given the data (the number of coin tosses and the number of heads occurring):

$$L(p|h, n) = \binom{n}{h} p^h (1-p)^{n-h}.$$

To maximize the likelihood, we vary the parameter (in this case the probability of a coin toss resulting in heads) and search for the value of the parameter that maximizes our

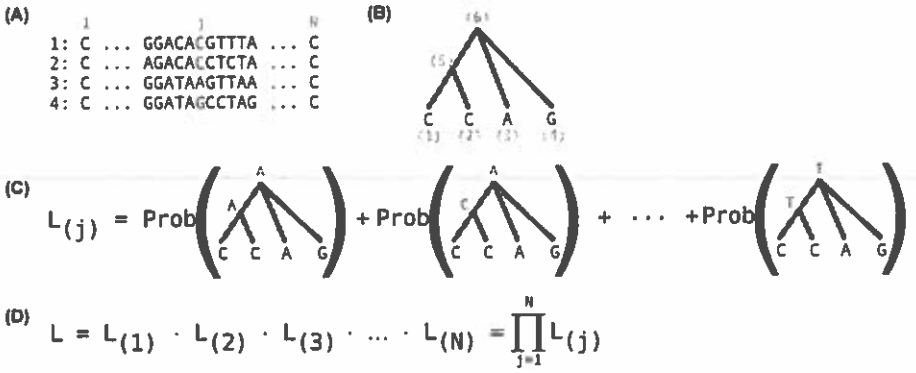


Figure 19: An illustration of the process involved in calculating the likelihood of a tree. (A) The sequences of four taxa, with sites 1 to N. Column j is the site we are using in this example. (B) The given rooted tree, (5) and (6) are the internal nodes of the tree to which we will assign nucleotides. (C) We calculate the likelihood for this site by summing the probability of each possible set of substitutions. (D) To calculate the likelihood over all sites we multiply the likelihood of each individual cite. Based on an illustration from [69].

likelihood. This procedure is nicely illustrated by plotting the likelihood L as a function of the probability of heads, p (fixing n and h according to the data). As shown in figure 18, we can see that the likelihood is maximized when the probability p is the proportion of heads that appeared during the experiment of n coin tosses (in this case 6 heads out of 10 coin tosses). We could also solve the problem analytically by taking the derivative of the likelihood function with respect to p and determining where the slope is 0 [35].

We will now explain how to extend the method to calculate the likelihood of a phylogenetic tree. As with the coin example, we need a set of data and a model. We will first describe the procedure for calculating the likelihood, and then we will discuss the inner-workings of the model. For now, consider the model a black box, where you give the model a tree along with a set of assignments of nucleotides to the internal nodes and it will return a probability of that particular tree occurring. Figure 19 (A) shows a common alignment of four homologous sequences with the sites in the sequences labeled 1 through N. Figure 19 (B) describes our tree with the nucleotides at site j of each sequence at the leaves of the tree. Internal nodes (5) and (6) have not been assigned nucleotides yet. To calculate the likelihood of this tree, we simply add up the probabilities of all possible assignments to the two internal nodes of the tree, as shown in figure 19 (C). In this case, with two internal nodes, there are four choices at each node and 2 nodes, so 16 possible nucleotide assignments to consider. Remember at this point, we have not discussed how the model is actually working, just that it is returning a probability for each nucleotide assignment we give it. After completing the summation of the probabilities of our 16 possible assignments, we now know the nucleotide assignment that gives the highest likelihood at site j. We then repeat this calculation for all sites in the sequences multiplying the resulting likelihoods together to get the final, aggregate likelihood for the entire tree. So, if our sequences are 256 nucleotides long, the likelihood of the tree over the entire sequence is $L = L_1 L_2 L_3 ... L_{256}$.

We will now consider the model itself and we will describe the model in its most general form (in the following section, we will discuss several specific models in more depth). Phy-

logenetic models specify the probability of observing certain site patterns and are generally expressed as a rate matrix

$$Q = \begin{pmatrix} - & \mu a \pi_c & \mu b \pi_a & \mu c \pi_g \\ \mu g \pi_t & - & \mu d \pi_a & \mu e \pi_g \\ \mu h \pi_t & \mu j \pi_c & - & \mu f \pi_g \\ \mu i \pi_t & \mu k \pi_c & \mu l \pi_a & - \end{pmatrix}.$$

Each element of the rate matrix specifies the probability of a substitution from nucleotide X to Y (the rate matrix would have 20x20 elements if it were describing amino acids and 61x61 elements if it were describing codons). Each component of the matrix has three parts: the background frequency, the transition probabilities, and the mean substitution rate. The background or equilibrium frequencies, denoted as π, describe the expected frequency of each nucleotide in a random sequence; the transition probabilities (a - l) represent the likelihood of element Y being substituted for element X and can be written as $P_{X \to Y}$. The final component of the rate matrix is the mean substitution rate of the model, which is related to the branch lengths in the tree. A branch length is defined by a rate of substitution μ and a time t. A branch may be long due to a high rate of mutation (μ) or because of a long passage of time (t) and it is not usually possible to tell what proportion each factor is responsible for. Models are usually specified with μ being combined into the rate matrix and t being expressed in the branch length, v, which we will discuss momentarily. The rate μ interacts with the specific transition probability for each nucleotide transition (a - l). Considering the Jukes and Cantor model, which assumes all nucleotides have equal background frequencies (such that $\pi_A = \pi_C = \pi_G = \pi_T = 0.25$) and assumes all transition rates are the same (i.e. $a = b = ... = l = 1$) we get a very simple model

$$Q = \begin{pmatrix} - & \frac{1}{4}\mu & \frac{1}{4}\mu & \frac{1}{4}\mu \\ \frac{1}{4}\mu & - & \frac{1}{4}\mu & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \frac{1}{4}\mu & - & \frac{1}{4}\mu \\ \frac{1}{4}\mu & \frac{1}{4}\mu & \frac{1}{4}\mu & - \end{pmatrix}.$$

There are many possible models that have been proposed to describe the evolutionary process and we will detail several of them in the following section [69].

So, as we left it above, our evolutionary model was a black box. In order to obtain the probability of a particular set of substitutions for a tree we perform the following calculations. We must have a tree, τ, a set of branch lengths for that tree, v_{XY}, connecting nodes X and Y and our rate matrix Q. If we have a simple tree, with two leaves and a root there is only a single internal node. We assign a nucleotide to our internal node, in this case, say we assign it a 'c', and that our leaves, A and B are known to have nucleotides 't' and 'g', respectively. The likelihood of this configuration of the tree, if we calculate from the root out to the leaves, is then: $L_c = \pi_c P_{c \to t} v_{RA} P_{c \to g} v_{RB}$, where π_c represents the probability of nucleotide C being there by chance, and P represents our rate matrix value [29, 35].

To calculate the tree we specified in figure 19 (B) for the site j with an 'A' assigned to the root and a 'C' assigned to the internal node, we would perform the following calculations (the nodes of the tree are labeled 1-6 in this case):

$$L_j = \pi_A P_{A \to G} v_{64} \pi_A P_{A \to A} v_{63} \pi_C P_{A \to C} v_{65} \pi_C P_{C \to C} v_{52} \pi_C P_{C \to C} v_{51}.$$

In our discussion of models, almost all our parameters were explicitly specified. In general, it is possible to use likelihood to optimize any of the parameters that belong to the

model. One case in point occurs with determining the proper branch lengths in a particular tree. One common approach is to perform a likelihood calculation on the smallest subtree that contains the branch. This calculation would be very similar to our example above where we optimized the parameter p, which specified the probability of heads in a coin tossing experiment, except in this case we would optimize the branch length v using our rate matrix Q as our model instead of the binomial distribution. A second approach to branch lengths is to build them directly into the rate matrix. So, instead of having a single matrix, you have several matrices each representing a common branch length [29].

So far we have assumed that a tree topology is given prior to calculating the likelihood. However, as we discussed in section 3.1.1, the number of possible trees rises extremely rapidly as we add more taxa to our analysis. One approach to identifying the topology would be to use a preliminary method, such as the neighbor-joining method to produce an initial tree topology. Another second method is to start with a minimum three species tree and then add species to it one at a time. For each new species, you calculate the tree configuration that gives the highest likelihood and then continue adding the next species [21]. A final approach is to search for the best topology by examining a certain fraction of the possible tree space. Any of the algorithms or heuristic methods described in section 3.1.1 can be applied here as well.

3.3.1 Alternative Evolutionary Models

One of the primary differences between phylogenetic methods described previously, such as parsimony or distance methods, and maximum likelihood, is that the former methods use implicit models of evolution. Parsimony favors the idea of minimum evolutionary change a priori [22], but does not specify the forces at work that yield minimum evolutionary changes to a tree. Distance methods generally use a measure of the difference of the number of substitutions in a set of sequences, which it then minimizes to cluster species, but again, the evolutionary forces that change the sequence data are not specified – sequence alignments are based on substitution matrices (see section 2.4), which are a reflection of several different models being applied to different and non-homologous branches in the tree [77]. Maximum likelihood, on the other hand, provides an explicit model for the data. In this section we will discuss these models in more depth starting with a simple example and then exploring a model of nucleotide change and a more complex model based on codon changes.

As mentioned in the previous section, to calculate the probability of observing a given site pattern several factors, including at least the transition probabilities (among nucleotides, codons, or amino acids) and background frequencies must be specified. Most likelihood models assume a time-homogeneous Poisson process to describe these transitions, which, among other things, implies that substitutions are independent and happen randomly. As a simple example of calculating transition probabilities, consider a two-state case, where the only character states are 0 and 1. The rate of change from 0 to 1, or 1 to 0, in a infinitesimal amount of time, δt is specified by a two dimensional rate matrix Q [35]:

$$Q = \{q_{ij}\} = \begin{pmatrix} -\lambda\pi_1 & \lambda\pi_1 \\ \lambda\pi_0 & -\lambda\pi_0 \end{pmatrix}$$

where λ is the rate of change from 0 to 1 or 1 to 0, and π_0 and π_1 are the equilibrium frequencies of each state.

To calculate the probability of observing a change over the interval of time t, we modify the rate matrix in the following way: $p_{ij}(t, \theta) = e^{Qt}$, where θ is a vector of parameters for the

model. This allows us to take our rate matrix of transition probabilities and express those probabilities as a function of time, which allows us to adjust for the fact that as a tree branch becomes longer, the probability of a transition having occurred rises. This calculation allows us to take a single set of transition probabilities and make them appropriate for a number of different branch lengths. Often these calculations can be performed analytically, such as for our hypothetical two-state example. However, if the model is complicated, the calculation can be done numerically [35].

In 1994, Ziheng Yang introduced the General Reversible Process Model (REV) as a means to describe nucleotide evolution [77]. REV is a Markov process that utilizes a four by four rate matrix Q to describe the probability that nucleotide i will be substituted with nucleotide j in an infinitesimal time interval δt. Similar to the generic model described above, π_A, π_C, π_G, and π_T describe the background frequency of the different nucleotides. In contrast to our general model, where we had 12 separate transition probabilities – one for every type of allowed substitution in the model, here there are only six transition rates, a, b, c, d, e, and f which are specified to make the model symmetric and provide for its reversibility. The rate matrix is specified as

$$Q = \begin{pmatrix} - & a\pi_c & b\pi_a & c\pi_g \\ a\pi_t & - & d\pi_a & e\pi_g \\ b\pi_t & d\pi_c & - & f\pi_g \\ c\pi_t & e\pi_c & f\pi_a & - \end{pmatrix}$$

The rate matrix Q is scaled so that time is represented as the number of substitutions per site and it may be converted, as described above, so that it can be applied to many different branch lengths.

One of the more interesting characteristics of Yang's model is that his general model encapsulates many of the more restrictive models that preceded it. If we assume that all four nucleotides have the same background frequencies and transition probabilities ($\pi = \frac{1}{4}$ and $a = b = ... = f = 1$), we get the model of Jukes and Cantor, first proposed in 1969. If we split our transition probabilities into two categories, including transition substitutions and transversion substitutions each with their own rates, we get Kimura's 1980 model. If we allow the four nucleotides to have different background frequencies we get Felsenstein's 1981 model. Finally, If we combine the special restrictions of different transition and transversion rates, along with different nucleotide frequencies, we get Hasegawa's 1985 model [77].

In the same year that Yang published his General Reversible Model, Goldman and Yang devised a model of nucleotide substitution based on modeling at the codon level [30]. The model uses both the nucleotide-level information in DNA sequences and amino acid-level information. For example, it incorporates into it the transition and transversion bias of nucleotides and the variability of a gene through information of synonymous and nonsynonymous substitutions (synonymous/nonsynonymous substitutions are discussed in depth in section 4.

As with the REV model, Goldman and Yang's model utilizes a Markov process to model substitutions among the codons within a protein-coding sequence. However, while the REV model used a discrete Markov process, the complexity of the codon-based model requires the use of a continuous-time Markov process. As a reminder, a codon represents three nucleotides which code for a particular amino acid in the DNA sequence.

The states of the Markov process are the 61 sense codons, the three stop codons are not included as they are assumed to largely change the gene, making their appearance very rare. This requires a 61x61 rate matrix, Q, to represent the probability of one codon changing

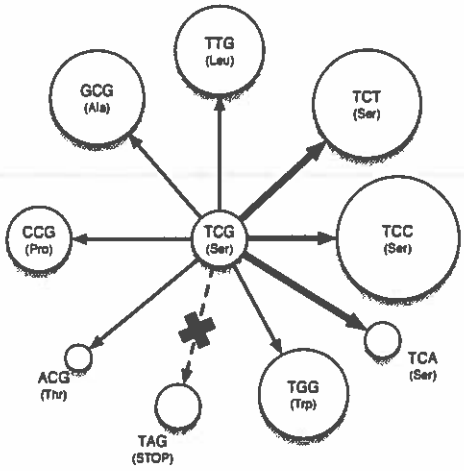


Figure 20: Each codon in the Goldman and Yang model may be converted into any of nine other codons if it experiences a nucleotide substitution. This figure shows an example for the 'TCG' codon. The size of the circles represents the frequency of that codon, transitions are marked as thick arrows, while transversions are thin arrows [30].

into another over a small amount of time. The model assumes that mutations occur at the three codon positions independently and only single substitutions are allowed. Each codon has at most nine neighboring states that it may change into (three nucleotides in each codon times three nucleotides that each could change into), an example of which is illustrated in figure 20. Rates of substitution involving a transition ($A \leftrightarrow G$ or $C \leftrightarrow T$) are multiplied by a factor κ, to allow for the ratio of transition and transversion substitutions to be incorporated into the model. In addition, to represent selective pressure, if a substitution in a codon modifies the amino acid, an additional multiplicative factor is incorporated, based on the physiochemical distances between the 20 amino acids and the general variability of the gene: $A = e^{-d_{aa_i,aa_j}/V}$ (the distance d_{aa_i,aa_j} is not taken from a substitution matrix, but rather is a simple integer that increases in size as the physiochemical distance between two amino acids grow, such as Grantham's index) [30]. This formulation expresses the fact that codons that code for chemically very different amino acids will rarely substitute for one another, while codons that code for chemically similar amino acids will substitute much more frequently. This matrix, although much larger, is still very similar in nature to the general model for nucleotides presented in the previous section. The parameters κ and A are calculated along with the existing model parameters including the background frequency, the transition probability, and the average mutation rate for each element in the rate matrix. Of course, these parameters now apply to the rates and frequencies of codons instead of nucleotides.

3.3.2 Among-site Rate Variation

One constant to the variety of models we have described so far is that the forces of evolution the models describe are assumed to affect all sites in a sequence equally. There may be a bias in the types of transition and transversion substitutions, but our models assume that bias

occurs equally at all sites. Evolutionary studies, however, have shown that substitution rate variation exists among sites in almost all genes. Different selective constraints at different sites are responsible for this conservation, usually because the particular functional or structural component being coded for is more or less important to the well-being of the organism [79, 78]. For example, estimated substitution rates at the first, second, and third codon positions can be ordered, $r_2 < r_1 < r_3$. The reason for this is the degeneracy of the genetic code. Changes to the third position in a codon very often do not change the amino acid it codes for, while changes to the second position always do. For this reason, genes accumulate more substitutions in the third position than in the second, but the models we have described so far do not account for it. Both Yang's nucleotide model as well as Goldman and Yang's codon-based model experienced problems due to among-site rate variation [77, 30].

If all sites in a sequence change at the same rate, the number of substitutions per site for a group of sequences should follow a Poisson distribution. However, Fitch and Margolish found, by examining the number of nucleotide changes in cytochrome c, that the data do not fit a Poisson distribution and that the data would only fit if a number of invariant sites (sites under strong selection) were excluded [79]. The hypothesis is that each site in the sequence has an unknown rate that is determined by its position in the resulting protein molecule; many portions of a protein-coding sequence may be completely invariant as they may code for an essential structural component. In the same way, other portions of the protein may be open to rapid substitutions due to a lack of constraints. As organisms diverge, a fast-changing site is assumed to experience substitutions at an elevated rate in all lineages, regardless of the particular organism-specific nucleotide at that site.

According to Yang, the standard approach to characterizing among-site rate variation is to use a statistical distribution, either discrete or continuous, to approximate rates at sites. For each site in the phylogenetic tree, one can sample from the distribution to determine its particular rate. The most commonly used continuous distribution for modeling the rate variation is the gamma distribution. The distribution is very flexible due to its shape parameter, α, and can either take on a bell-shaped curve or an L-shaped curve depending on whether or not α is greater or less than one, respectively. Several examples of the gamma distribution are illustrated on the left side of figure 21.

While sampling from a continuous distribution is thought to be the most accurate method to estimate rate variations at different sites, it is computationally intensive. A practical approach is to instead use a discrete-gamma model, in which several classes of rates are used to approximate the continuous gamma distribution. The classes are chosen so that each class has an equal probability of occurring. An illustration of a discretized gamma distribution with eight classes is shown on the right side of figure 21. The value of α can be added as an additional parameter to our likelihood model and its value can be estimated from the data [78, 79].

3.3.3 Model Comparison

One characteristic of phylogenetic likelihood analysis that makes it attractive is the ability to compare one model to another by using a Likelihood ratio test. Calculating the likelihood ratio allows a comparison between the relative tenability of two hypothesis. If L_0 specifies the likelihood of the null hypothesis and L_1 specifies the likelihood of the same data under

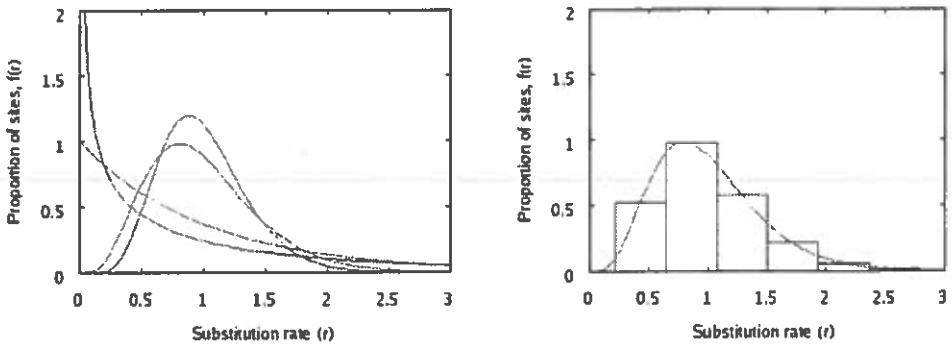


Figure 21: On the left, several examples of the gamma distribution with different values for the shape parameter, α. When α is less than one, an L-shaped distribution appears, when α is greater than one, a bell-shaped distribution occurs. On the right side is an example of a discretized gamma distribution with $k = 8$ classes.

a different hypothesis then the ratio can be calculated by

$$\Lambda = \frac{max(L_0(\text{Null Model} \mid \text{Data}))}{max(L_1(\text{Alt model} \mid \text{Data}))}.$$

When Λ is less than one the null hypothesis is discredited, when it is greater than one the alternative hypothesis is discredited. One method to generate the null hypothesis is through parametric bootstrapping, which we will discuss in depth in section 3.3.4 [35].

3.3.4 Bootstrapping

We have now spent considerable time discussing how to determine the likelihood of a given tree as well as how to search the space of tree topologies to maximize our likelihood. We would like to be able to specify a level of confidence in these results, to measure the accuracy of the values we optimized for the parameters in our model. The bootstrap is a method that involves randomly resampling from our own data set to determine empirically the variability in our estimation [23, 20].

Suppose you had n data points $x_1, x_2, ..., x_n$ which were drawn independently from the same distribution along with a parameter t whose value is being optimized from the data via a statistical estimate function T. If the exact distribution of x_i were known we could derive formulas for the standard error and calculate confidence intervals for t. The bootstrap procedure is useful when you do not know the distribution from which x_i is drawn. The central idea of bootstrapping is that if the original sample size n is large, each possible value x_i will be represented in the sampled data at the same proportion as in the true underlying distribution; and randomly resampling our sample data will be the same as sampling from the underlying distribution [23].

In order to apply bootstrapping here, we resample from our data set $x_1, ..., x_n$ to construct a series of randomized data sets. Some of the data points will be represented more than once, others not at all. For each fictional set of data, we then optimize our parameter t:

$$t^* = T(x_1^*, x_2^*, ..., x_n^*)$$

After repeating this procedure r times, we have a collection of r estimates of our statistic T. The actual estimate t and the variance and confidence limits for t can now be inferred by computing them for t^*.

Applying the bootstrapping technique to phylogenetic data starts with a multiple sequence alignment, X, consisting of the sequences that will be used to form the phylogenetic tree. After building the tree, the bootstrapping method proceeds by taking the matrix X and selecting columns from it randomly to form a new matrix X^*. This is done with replacement, so column 14 in X may become column 37 in X^*. Next, the tree is again constructed, this time based on the data from X^*.

This process is repeated some number of times and the proportion of bootstrap trees that agree with the original tree in their topology are calculated. A confidence score is for each branch in the tree is then formed: if a clade appears in the same location in 193 out or 200 bootstraps than the branch has a confidence value of 0.965 [20].

3.3.5 Maximum Likelihood Consistency and Limitations

Conflict between proponents of parsimony and maximum likelihood estimation are well known. Many of the arguments are based on philosophical disputes regarding the superiority of explicit versus implicit models, with many of the arguments centering around unprovable assertions about the proper way to perform inductive inference in science. For example, which is better, Ockham's razor or Fisher's likelihood principle? Nonetheless, one of the major contributions that maximum likelihood estimation brought was the idea that if we are to compare and contrast phylogenetic methods we must do so through a common framework of statistics; it carries a set of standard results and tools that allow systematic investigation of its properties [65, 22].

As with parsimony and distance methods, there are certain cases in which maximum likelihood estimation is inconsistent. Recall that an estimator is consistent if it converges to the true parameter as data accumulates. Recently, maximum likelihood was proven to be consistent for the general model we described above, but has not been proven for some popular models, such as models accounting for site variation using a gamma distribution. Being consistent often relies on whether tree topologies are identifiable. A tree topology is not identifiable if different trees can generate the same sampling distribution of the datasets. In this case, no method can correctly estimate the right tree, regardless of sample size. This property often emerges when models are very complex, especially when rate variation across sites is allowed [65].

An additional source of inconsistency in maximum likelihood estimation occurs when the rates at which sequence sites evolve change asynchronously over time [46]. While we have discussed above how to incorporate different rates of substitution among different sites in a phylogenetic tree, often, functional constraints on a specific site change over time. This means, not only do rates vary among sites, but rates change at specific sites over time. This phenomenon is known as heterotachy. When an identically distributed evolutionary framework, such as a maximum likelihood model, is imposed on sequences that evolve heterogeneously, parameter estimates are compromises over sites and lineages and are therefore incorrect for many sites and can lead to inconsistency. While it is possible that heterotachy can be handled with additional parameters in the model, similar to the way a discrete gamma distribution is used to handle site variations, there is not yet a model that accounts for this issue [46].

One of the main arguments employed in favoring maximum likelihood methods is their

ability to incorporate knowledge of molecular evolution through the use of models. Therefore, one of the following should be true:

1. If the model is important, there will be a relationship between the validity of the model and the accuracy of tree construction.

2. If model validity is not that important, then that implies that the signal of evolution can be detected under a variety of models and different methods.

However, there exists a large number of models for maximum likelihood methods, with varying numbers of parameters and no clear consensus surrounding them. Consider the number of possible models to choose from, which increases with the number of parameters. For K parameters, the number of models is given by the number of possible subsets of K distinct elements. Considering the general model we described above, there are 17 parameters in the rate matrix alone (4 frequencies, 12 transitions, and the mean mutation rate). In addition, we could add a parameters for site rate variation (based on the number of discrete gamma categories we want), as well as possible branch length parameters. In Goldman and Yang's codon model, we easily approach 70 parameters (4 frequencies, 61 transitions, the mean mutation rate, amino acid similarity measures, transition/transversion bias, etc.)!

One final issue that is of great importance regarding maximum likelihood estimation is computational complexity. Maximum likelihood methods consist of an outer loop that searches among trees, and an inner loop that calculates the likelihood of a particular tree. Searching for the optimal tree is known to be NP-hard with the time required to find the tree increasing exponentially as a function of the number of species in the tree [65]. Depending on the number of parameters that are fixed in our model (frequencies and transitions), versus those that have to be optimized (possibly branch length and site rate variation among others) we can quickly create a estimation problem that is intractable. In fact, our ability to estimate phylogenetic trees, with more accurate, parametric models, along with additional species data will remain bounded by a lack of computational resources for some time to come [65].

3.4 Bayesian Methods

The fourth and final major type of phylogenetic analysis we will examine is Bayesian Inference. Bayesian estimation is one of the oldest methods of statistical inference, dating back to the 18th century. Like the maximum likelihood method, Bayesian estimation of phylogeny utilizes an evolutionary model, which determines the likelihood of observing a set of data conditional on a particular tree and a set of parameters, and the same models can be used in both techniques. In contrast to maximum likelihood, Bayesian analysis treats model parameters in two significantly different ways. First, likelihood methods assume that each parameter has one true value and it attempts to find that value by searching the entire space of the parameter. Bayesian analysis assumes that parameters have no single true value, but each parameter has a distribution. Secondly, Bayesian analysis asserts that, before the data are observed, parameters have a prior distribution. This prior distribution incorporates previous information known about the parameters and it is combined with the evolutionary model to produce the posterior distribution. All inferences about the model or the data are then made with the posterior distribution. While we will explain it in more depth below, figure 22 (A) may provide some intuition as to how Bayesian inference works [36, 37, 80].

The Bayesian method is attractive because it formalizes the natural scientific method where a scientist starts out with some beliefs about a hypothesis, signified by the incorporation of prior probabilities, collects data, and then modifies his or her beliefs respective to the observations. Let us return to the problem of determining the fairness of a coin. Consider that you are asked to bet money on the relative probability of turning up heads after tossing a particular coin. You are only allowed to toss the coin twice and heads comes up both times. Based on these observed data, a maximum likelihood analysis would give an estimated probability of heads as 100%. Although it is possible that the coin is highly biased, it is more likely given what we already know about coin tossing that if the coin were fair, the probability would be around one-half. Bayesian analysis allows you to combine this 'background' information in the estimation via the prior distribution [36].

In detail, Bayesian inference works in the following way. To calculate the posterior probability of a tree, we need four components: the set of possible trees for s species, $B(s)$; a model, which we will refer to by its set of parameters, θ; a set of branch lengths for each tree, v; and a set of data observations – an alignment of homologous DNA sequences, which is referred to as D. The ith tree in $B(s)$ is referred to as τ_i. The posterior probability of trees is the probability of the ith tree conditional on the observations and is calculated using Bayes theorem. Bayes theorem multiplies the prior probability of the tree with the likelihood of the observations given our tree and divides by a normalizing constant that involves summation over all trees:

$$f(\tau_i|X) = \frac{f(\tau_i)f(X|\tau_i)}{\sum_{j=1}^{B(s)} f(X|\tau_j)f(\tau_j)}.$$

To calculate the probability of a particular tree in the previous formula we must integrate over all possible combinations of branch lengths and substitution model parameters multiplied by the prior distribution of the branch lengths and model parameters:

$$f(\tau_i|D) = \int_v \int_\theta f(D|\tau_i, v, \theta) f(v, \theta) dv d\theta.$$

This last step, integrating over all possible parameter values is one major advantage of likelihood methods. Likelihood methods may only select a single value for each parameter and these choices may be incorrect. Bayesian inference allows us to look at the distribution of all parameters.

Let us consider a simplified example as presented by Yang in [80]. Say we have two nucleotide sequences where x of the n sites in the sequences are different and we are using the Jukes-Cantor model of evolution (equal background frequencies of the nucleotides as well as equal transition rates between nucleotides). We want to estimate a single parameter, θ, from our data which represents the distance between the two sequences as the expected number of nucleotide substitutions per site. The probability that a site is different between two sequences is then

$$p = \frac{3}{4}(1 - e^{(-4/3)\theta})$$

. The likelihood of observing x differences out of n sites is then given by the binomial probability

$$L(\theta|x) = f(x|\theta) = Cp^x(1-p)^{n-x}$$

where C is a constant. Finally, we will use an exponential distribution as our prior

$$f(\theta|\mu) = \frac{1}{\mu} e^{-(1/\mu)\theta}.$$

Using an exponential distribution as our prior weights θ towards smaller values (since the vast majority of the area under the curve is close to 0. It says that our prior beliefs tell us that we expect small values for θ. We can now determine the posterior distribution using Bayes theorem

$$\frac{f(\theta|\mu)f(x|\theta)}{\int f(\theta|\mu)f(x|\theta)}.$$

The results of this calculation is plotted in figure 22 (A). It demonstrates how the prior distribution affects the posterior – the posterior distribution is pulled off the peak suggested by the likelihood distribution.

Returning to the more complicated subject of fully parametrized models and trees, we can infer phylogenetic conclusions from the posterior probability distribution. One way to do this is to use the most probable tree as a point estimate of phylogeny, known as a maximum a posteriori probability (MAP) estimate of phylogeny. A second use of the posterior probability distribution would be to construct a 95% credibility interval for a set of trees (a credibility interval is similar to, but not the same as, a confidence interval). This is done by starting with the MAP tree and continuing to add the most probable trees until the cumulative probability reaches 0.95.

As mentioned above, another possibility, and perhaps the most powerful, is to summarize the results of the Bayesian analysis on a majority-rule consensus tree. This method is similar to a bootstrap analysis (bootstrap analysis is discussed in section 3.3.4), where we can use the sampled set of trees to determine how many of them agree on the same branches. The main difference is that the numbers on the tree branches represent the posterior probability that the clade is true.

3.4.1 Approximating the Posterior Probabilities of Trees

Calculating the posterior probability involves summing all possible trees and, for each tree, integrating over all parameters for that tree. This calculation is not analytically possible and must therefore be approximated. Markov chain Monte Carlo (MCMC), is used as the approximation method in Bayesian inference and can be thought of as a sampling procedure. While we can't analytically determine the full parameter and tree space, if we take enough samples from it, we should be able to approximate the underlying distribution. This property of Bayesian inference makes it much more efficient than maximum likelihood, where the entire parameter space is calculated.

This process can be thought of using an analogy of a wanderer climbing a hill. The wanderer considers taking a random step from his current location in a random direction. If the step is uphill, the wanderer chooses to take the step unconditionally. However, if the step is downhill, it is not rejected outright, but is taken based on a certain probability. After a long period of time, the wanderer will have explored the hill thoroughly and will have spent time in each location proportional to the height of that location (more time in the higher elevations) [80].

We return once again to our simple Jukes-Cantor substitution model, where we are trying to estimate the value of a single parameter θ, which represents the distance between two aligned sequences. Figure 22 (B) shows three chains, each started from a different location. After a certain amount of time, each chain converges on a certain range of the parameter θ. If you compare the region where the chains occupy their time in figure 22 (B) with the likelihood curve in figure 22 (A) it is clear that the chains converge on a value of θ that maximizes the likelihood. As an aside, it takes a certain amount of time for each chain to

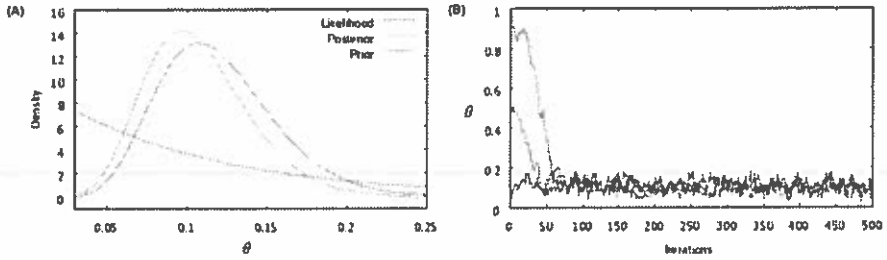


Figure 22: (A) The prior, posterior, and likelihood distributions showing the optimal value of θ given the data. (B) An example of Markov chain Monte Carlo. Based on an example from [80].

converge – approximately 100 iterations of the chain. This period is known as the 'burn-in' period for the Markov chain.

MCMC, as applied to phylogenetic trees works in the following way. First, you start with an initial tree – the tree can be randomly chosen, or it can be one that is a known, reasonable approximation of the true tree. A new tree is then proposed. The mechanism that proposes the tree must be stochastic, and every possible tree must be accessible by the mechanism. Also, proposed trees must be aperiodic. A number of different proposal mechanisms might be used, each of which changes a different parameter, such as branch lengths or the tree topology, or the parameters of the likelihood model. The idea of the proposal mechanism is to be able to create a representative sample of the entire tree space.

The new tree is accepted or rejected based on a probability. That probability is based on the multiplication of three ratios, the likelihood ratio, the prior ratio, and the proposal ratio, each of which compares the new model value against the old for that particular ratio. So, for example, the likelihood ratio compares the new proposed tree, given the data, to the old tree given the data. This value is known as R.

Finally, a random number is generated on the interval of (0,1). If that number is greater than R, than the new tree is accepted, otherwise it is discarded. This process is repeated thousands or millions of times: the fraction of the time that the chain visits any particular tree is a valid approximation of the posterior probability of that tree [37, 38].

3.4.2 Chain Convergence

There is a 'burn-in' period that occurs before the MCMC algorithm can converge from a random start to a set of parameters that begins to approximate the posterior distribution. To determine if the MCMC algorithm is indeed converging, one can execute more than a single chain. If after some time, both chains converge on the same parameter values, despite being randomly started in different parameter spaces, then you can be sure the method is converging.

One method that improves convergence is the Metropolis-coupling technique. Using this method, one has n Markov chains. One of those chains samples from the posterior distribution of interest and is known as the cold chain. The other chains are sampling from a 'heated distribution', which is obtained by taking the cold distribution and multiplying it by a large value (greatly perturbing the chain). At regular intervals, two chains are picked at random and their states are exchanged as in a regular MCMC step.

The effect of using heated chains is to decrease the distance between local maxima in the posterior distribution. This allows the heated chains to move more freely between isolated hilltops. The only function of the heated chains is to supply intelligent proposals for new states allowing the cold chain to jump from one hill to another in a single step. In normal conditions, traveling between hilltops could take millions of iterations. This optimization is known as Metropolis-coupled MCMC, or MCMCMC (MC^3) [37]. This optimization has allowed one to integrate over a tree space that is several hundred orders of magnitude larger than conventional MCMC spaces.

3.5 Conclusions

If sequence alignment could be considered a method that relies on the bredth of genetic data, than phylogenetic inferrence is a method that focuses on the depth of genetic data. Of the four major types of phylogenetic methods we examined in this section, two of them are being very heavily researched. While parsimony and the distance methods are still used in some algorithms, they are generally used to generate rough or initial versions of trees that will be further refined through another method. One example we have seen is with the program CLUSTALW, which uses the neighbor-joining distance method to produce its initial guide tree. Many of the algorithms to enumerate trees, while developed intially for use with parsimony algorithms, are now being widely used to enumerate trees in maximum likelihood algorithms. Major research continues in the development of new and better models – for use in both maximum likelihood and Bayesian algorithms. Finally, there is a lot of active work in the Bayesian realm, much of it focusing on the characteristics of the sampling space of trees and model parameters.

If we look beyond the algorithms themselves and examine the application of phylogenetics, one exciting area involves the ressurection of ancestral genes [73]. In these analysis, researchers are using phylogenetics to infer the ancestral state of certain genes, such as genes that encode steroid hormone receptors, and they are then ressurecting these ancient genes in the laboratory to study their ancient functions! In terms of whole genome duplication, phylogenetics is playing a small, but increasing role in verifying the occurances of duplication topologies. There are two major problems that have prevented researchers from using phylogenetics more widely in these types of analyses. First, identifying the signal of whole genome duplications is very resource-intensive and phylogenetic algorithms can consume extremely large amounts of computational power. Second, the evolutionary models used by phylogenetic algorithms are heavily parameterized and finding the right combination of parameters, that can be applied across many groups of genes in many species is a difficult task. Applying phylogenetics to the study of whole genome duplication can only be expected to grow, however, particularly as models improve and Moore's law continues on its course.

4 Genetic Drift and Synonymous/Nonsynonymous Mutation Rates

In this section we will focus on the theory of genetic drift and one of its major implications: a measurement of how powerfully natural selection is acting on a set of related genes. The neutral theory of evolution, also known as genetic drift, describes the effects of random mutation and the fixation or loss of those mutations in a population that is not under the influence of natural selection. By understanding how mutations function in a neutral envi-

ronment, one in which natural selection is not active, we can infer the presence of positive or negative selection based on how the mutations we observe deviate from the behavior predicted by the neutral theory. This deviation can be quantified by measuring the synonymous/nonsynonymous mutation rate. These measurements have been used to hypothesise about the existence of molecular clocks; an idea that supposes that the rates of change of nucleotides are relatively fixed allowing us to calculate the ages of inferred changes in genes. In terms of genome duplication, measuring the synonymous/nonsynonymous mutation rates in duplicated genes may provide us with information as to the fate of duplicated genes. Are duplicated genes undergoing increased rates of change after a duplication event? We will proceed in this section by first describing the neutral theory, as well as a revision known as the nearly-neutral theory, followed by two methods to measure synonymous and nonsynonymous mutations.

4.1 The Neutral Theory of Evolution

In the middle of the twentieth century the prevailing view of evolution was that change at the morpological (the shape and structure of an organism) and functional levels resulted from the process of natural selection operating via adaptive changes to DNA sequences. Further, it was widely believed that selection for fitness was primarily a top-down, hierarchical procedure. According to this hypothesis, there was cellular control of molecular activities, organismal control of cellular activities, and populational control of organismal activities; the core idea was that DNA was a passive carrier of the evolutionary message and that positive selection drove mutations and the mutation rate to increase fitness [45, 44]. Evolutionary biologists who believed in this view were known as selectionists.

In 1969, Jack King and Thomas Jukes proposed a hypothesis that inverted the selectionist viewpoint (Motoo Kimura proposed a similar hypothesis in another paper at roughly the same time). King and Jukes asserted the view that most evolutionary change to DNA was not due to Darwinian natural selection and, further, that the forces of evolution provided for no hierarchical control of organisms or their populations. That is to say that a population of organisms had no control of the evolutionary changes to its individual members in the same way that individual organisms had no control of the direction of mutation going on within their own cells. Put another way, change observed at the phenotypic level does not necessarily apply at the genotypic and molecular levels (the geneotype of an organism is its genetic specification, i.e. a series of genes encoded in DNA, while a phenotype is the physical organism that results from that genetic specification). The central idea of the neutral theory is that evolutionary change is not imposed upon DNA from without, but instead it arises from within. At the molecular level, random genetic mutations, which have no effect upon the fitness of an organism, occur and can become passively fixed in a population through the action of random genetic drift.

As we saw in section 1.2, the genetic code for amino acids is degenerate. That is to say that amino acids are specified by codons, which themselves are made up of nucleotide triplets. Because there are four choices for each of the three nucleotide positions, there are 4^3 possible codons. However, since there are only twenty amino acids used to specify proteins multiple codons can specify the same amino acid. Given any particular codon, it can mutate in any of nine ways by a single nucleotide substitution (this process is illustrated in figure 20). When a nucleotide mutates within a codon but does not change the resulting amino acid it is called a *synonymous* substitution. Likewise, nucleotide mutations that do change the resulting amino acid are called *nonsynonymous* substitutions.

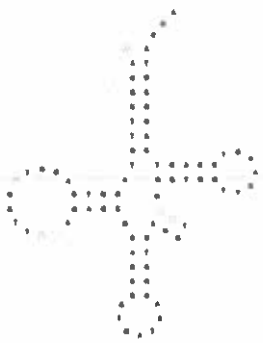


Figure 23: The transfer RNA molecule can accomodate a small number of changes to the amino acids in the helical regions without affecting its functional properties. Such changes are known as nonsynonymous mutations and may become fixed through genetic drift.

Of the 549 possible single-nucleotide substitutions, 134 (25%) of them are substitutions to synonymous codons [45]. So, it is therefore possible for a mutation to occur in one of the three nucleotides for a particular codon and for the amino acid it codes for to remain the same – making the mutation completely invisible to the forces of natural selection.

While synonymous substitutions are truly invisible to the organism and natural selection, other substitutions do result in altered proteins, but are still neutral with respect to organismal function and natural selection. Protein molecules are continually challenged by mutations in the DNA that codes for particular amino acids. These changes are screened by natural selection. Those mutant substitutions that disrupt less the existing structure and function of a molecule occur more frequently in evolution than more disruptive ones. For this reason, substitutions are more likely to be synonymous, and if not synonymous, they are likely to code for an amino acid that is chemically similar to the one being replaced. Moreover, mutation rates differ not only between different protein molecules, but also between different parts of one molecule.

In general, as long as the nucleotide substitutions do not radically impair the function of a protein, by changing certain invariant sites, than those changes may become fixed. Sites may be invariant because they are necessary to fulfill an enzymatic function of the protein, or, changes at invariant sites may affect the secondary or tertiary structure of the protein, by interfering with the proper folding of the protein. For example, the helical regions of the transfer RNA (tRNA) molecule seem to be able to accommodate a few changes in the base pairings without disruption of function, but it was empirically determined that the regions are restricted to no more than one or two such changes. The empirical evidence suggests that additional mutations in the helical region become highly deleterious and are rejected by selection [44].

The existence of neutral mutations makes it possible to resolve a dilemma in the evolution of primates and guinea pigs. Both of these organisms are unable to properly produce ascorbic acid and are subject to scurvy when on a diet that lacks vitamin C. All other animals are free of this restriction and are not subject to scurvy providing the question of how did such a nonadaptive change occur under a pure natural selection theory? King and Jukes proposed that since a diet lacking vitamin C was rare among primates, that the loss of the ability to

produce ascorbic acid was a neutral mutation that became fixed in the population [45].

4.1.1 The Nearly-Neutral Theory of Evolution

According to the neutral-mutation, random drift theory advocated by Kimura, King, and Jukes most evolutionary changes at the molecular level are caused by random genetic drift of selectively neutral mutations, rather than by natural selection. Almost two decades later, Tomoko Ohta, who was Kimura's student, proposed a revision of the theory to clarify the interaction of natural selection and random drift. His theory stated that natural selection is not an all-or-nothing force, and that there are many types of mutations whose behavior is influenced by both selection and random drift [60].

With the neutral theory, mutations occurred and were either synonymous, in which case they were invisible to natural selection, or they altered the protein molecule in a non-deleterious way. Natural selection only participated if a mutation occurred in a conserved, or invariant region, in which case purifying selection would remove the mutation from the population or, in extremely rare cases, if a mutation substantially increased fitness (usually due to environmental changes). Ohta's primary contribution was to add a third class of mutations – those that are *nearly* neutral. Mutations in the nearly neutral class would be affected by both selection and drift. The two theories are illustrated in figure 24 (A).

According to Ohta, it is important to differentiate between mutations and evolutionary substitutions. Numerous mutations appear every generation, but the majority of them will be lost within a few generations. Evolutionary substitutions, however, become fixed in the population. A mutation may be fixed by the selective advantage it confers to its host, or, it may be fixed simply by genetic drift. A large number of mutations, however, will be in between the two classes and these comprise the nearly neutral mutations. These mutations might be slightly deleterious, but not damaging enough to trigger full purifying selection, or, they might be slightly advantageous, but not enough to trigger strong positive selection. This relationship is plotted in figure 24 (B). The plot represents the fixation probability, u, of a mutation as a function of the product of the population size and selection coefficient, Ns. p represents the initial fraction of the population that has the mutation and the line $u = p$ represents the fixation probability of a neutral mutation. The probability of a mutation being fixed or purified by selection is controlled by the size of the population (N) and the strength of selection s. Near-neutrality occurs when the selection coefficient, s, approaches zero.

Finally, large adaptive changes at the molecular level in higher organisms would then be seen not through directed, positive selection but through chromosomal changes such as duplication and illegitimate crossing-over of chromosomal segments. These forces culminate in the creation of multigene families, which then evolve in concert with one another [60].

4.1.2 Molecular Clocks

The subject of molecular clocks is large and varied and is outside the scope of this document. However, suffice it to say that the basic idea of a molecular clock is to use the rate of neutral mutations to measure how long ago various evolutionary events occurred.

Kimura and Ohta argued that the rate of evolution, with regard to neutral nucleotide substitutions is approximately constant per year per site for various organisms. Based on studies of α and β chains of hemoglobins, Kimura suggested that the rate of substitutions is approximately 10^{-9} per site per year. Although there are local fluctuations, when averaged over a long time period, Kimura found the rate of evolution to be remarkably uniform

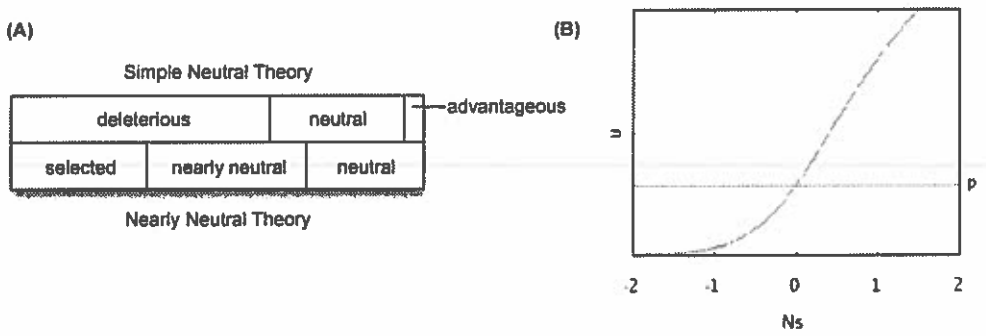


Figure 24: (A) A comparison of the simple neutral theory with the nearly-neutral theory, from [60]. (B) The fixation probability u of a mutant as a function of the product of the population size and selection coefficient, Ns.

among different lineages. The idea of a molecular clock becomes more complicated when other factors are considered. For example, Kimura's clock is based on substitutions per year and utilizes amino acid substitutions. Ohta, on the other hand, found that the substitution rate was not constant per year when using nucleotide data, but was instead affected by the generation length of the particular organism being studied. Further, we have seen in section 3.3.5 that the mutation rate at different sites in different molecules can change over time in a phenomenon known as heterotachy.

4.2 Estimating Synonymous and Nonsynonymous Substitution Rates

With the neutral theory as a foundation or null model, we will present here a method to estimate the number of synonymous (S) and nonsynonymous (N) sites in a pair of protein coding genes. This information will enable us to determine the *rate* of nonsynonymous (d_N) and synonymous (d_S) substitutions in those genes and further, comparing these rates as a ratio $\omega = d_N/d_S$ will allow us to make inferences about the strength and direction of natural selection. For example, if the rate of nonsynonymous substitutions is equal to the rate of synonymous substitutions, with $\omega = 1$, then selection is not acting on the genetic seqeunces we are examining. We know this since the rate of substitution is equivalent to the background rate or, the rate the neutral theory predicts as genetic drift. If however, $\omega < 1$, than that would imply that purifying selection is working to remove deleterious effects from a gene. Likewise, if $\omega > 1$, it implies that positive selection is in effect, proliferating genetic changes that confer some advantage in fitness. We will examine two algorithms to compute these rates, first examining the method proposed by Motoo Kimura and modified by Wen-Hsiung Li, and then Ziheng Yang's method involving maximum likelihood. Both methods have three basic steps: tabulate the number of synonymous and nonsynonymous substitutions in two genetic sequences, count the differences in synonymous and nonsynonymous sites between the sequences and, finally, correct for multiple substitutions at the same sites [81].

Li's method begins with the classification of nucleotide sites within a codon. As shown in figure 25 (A), a site can be classified as either fourfold degenerate, twofold degenerate, or nondegenerate. If there is a substitution at a fourfold degenerate site within a codon, it does not matter which nucleotide is substituted, since in every case the amino acid that

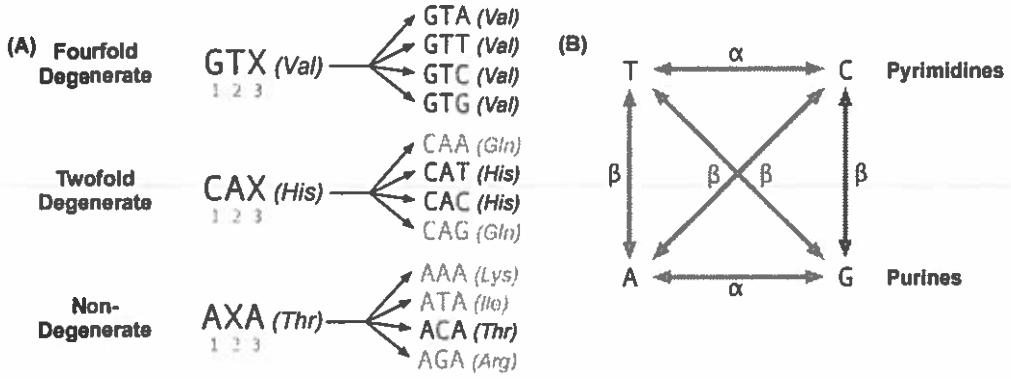


Figure 25: (A) Classification of nucleotide sites within a codon. The degeneracy of a site refers to the number of substitutions that will cause a change in the codon; in a fourfould degenerate site no substitution will change the codon, in a twofold degenerate site two substitutions will change the codon, and in a nondegenerate site any substitution will change the codon. (B) Types of evolutionary substitutions and their rates according to Kimura's model in [43].

the codon specifies will not be changed – all three substitutions are synonymous. In the case of twofold degenerate sites, one of the three possible nucleotide substitutions will not change the codon, but the other two will, and in the case of a nondegenerate site, any change to the nucleotide will change the codon and its resulting amino acid – a nonsynonymous substitution. By far, the third position in a codon is the most susceptable to synonymous changes, with 32 of the 61 sense codons falling into this category. On the other extreme, the second position of all sense codons is nondegenerate and any substitutions in these locations will change the codon and the amino acid it specifies [52]. What these numbers imply is that the third codon position is most likely to accumulate substitutions over time since synonymous substitutions are invisible to selection while the second position is least likely since changes to amino acids are most often deleterious making the mutation the subject of purifying selection (with the first codon position falling in the middle of the other two).

Once all the sites have been classified, the numbers of occurances of each type is summed: L_0, L_2, and L_4 representing the average number of nondegenerate, twofold and fourfold degenerate sites respectively per sequence. We then average the number of degenerate sites between sequences to account for the fact that corresponding codons from the two sequences may have different numbers of degenerate sites. For example, if the first sequence contains the codon CCG, which specifies proline, it has nondegenerate nucleotides in its first two positions and a fourfold degenerate nucleotide in the third position. If the corresponding codon in the second sequence was CGG, which specifies arginine, it has a twofold generate nucleotide in its first position, a nondegenerate nucleotide in its second position, and a fourfold degenerate nucleotide in its third position (this particular example is illustrated as the last set of codons in figure 26 (A)). Differentiating our sites by the type of degeneracy is necessary in order to properly compute the synonymous and nonsynonymous rates. The reason is that nondegenerate sites are nonsynonymous, fourfould sites are synonymous, but twofold sites are sometimes synonymous and sometimes not. When the substitution at a twofold degenerate site is a transition, or a change from a purine to a purine or a pyrimidine to a pyrimidine, it is synonymous, but when the substitution is a transversion, or a change

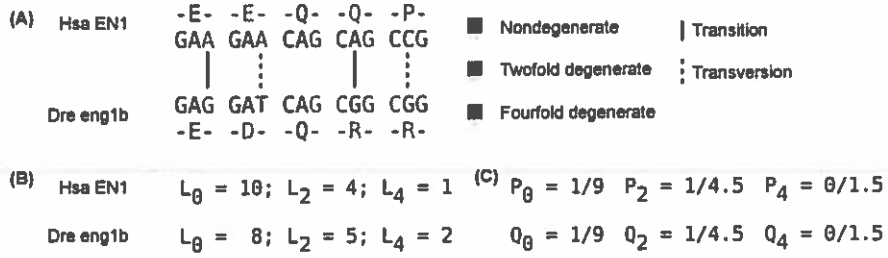


Figure 26: An illustration of the first three steps in Li's ([52]) method to estimate synonymous and nonsynonymous substitution rates using sequence data from the human engrailed gene (*EN1*) and the zebrafish ortholog (*eng1a*). (A) Classifying the nucleotide sites in each seqeunce and determining the types of observed substitutions. (B) Tallying up the nucleotide site types as nondegenerate (L_0), twofold degenerate (L_2) and fourfold degenerate (L_4). (C) Calculating the number of observed transitional (P_i) and transversional (Q_i) differences in the sequences.

from purine to pyrimidine or pyrimidine to purine, it is nonsynonymous [10].

Next, the two sequences are compared codon by codon in order to tabulate the number of observed changes with p_i representing the number of transitions and q_i representing the number of transversions for each type of site ($i = 0, 2, 4$). We then specify a ratio of the number of observed changes to the number of expected changes for transitional (P) and transversional (Q) changes:

$$P_i = \frac{p_i}{L_i}$$
$$Q_i = \frac{q_i}{L_i}.$$

These processes are illustrated with an example from the human engrailed gene (*EN1*) and the zebrafish ortholog (*eng1a*) in figure 26.

The final stage in the algorithm is to apply Kimura's two-parameter method to actually esitmate the rates of synonymous and nonsynonymous substitution [43]. The model that underlie's Kimura's method is illustrated in figure 25 (B). For each of the four nucleotides it describes the types of substitutions that might occur as being either a transitional or transversional changes and assigns a rate for each of the two types of changes: α representing the rate of transitions and β representing the rate of transversions. As figure 25 (B) describes, there are four possible transitional changes and eight possible transversional changes and therefore the total number of changes over a small period of time would be $k = \alpha + 2\beta$. Given the probabilities of P and Q, which we calculated above, as well as a time T, Kimura then derives a pair of differential equations to describe the number of mutataions that will occur at time $T + \delta T$. These equations can be solved analytically to calculate the α and β rates of substitution (for a full derivation of these equations see [43]). Li applies these equations to first calculate the number of transitional (A_i) and transversional (B_i) nondegenerate, twofold and fourfold degenerate sites:

$$A_i = (1/2)\ln\left(\frac{1}{1-2P_i-Q_i}\right) - (1/4)\ln\left(\frac{1}{1-2Q_i}\right)$$
$$B_i = (1/2)\ln\left(\frac{1}{1-2Q_i}\right)$$

and then the total number of substitutions per nondegenerate, twofold, and fourfold degenerate sites is given by:

$$K_i + A_i + B_i.$$

Finally, he calculates the overall rates of synonymous substitutions per synonymous site (K_S) and nonsynonymous substitutions per nonsynonymous site (K_A):

$$K_S = \frac{3(L_2A_2 + L_4K_4)}{L_2 + 3L_4}$$

$$K_A = \frac{3(L_2B_2 + L_0K_0)}{2L_2 + 3L_0}.$$

There is one major component of a full solution to synonymous rates that has not been discussed yet. When there is more than a single change in a pair of codons we run into a problem of establishing the proper order of the changes to the codons. For example, if we were to have continued on with the example in figure 26 (A), the next pair of codons in the human and zebrafish engrailed gene would be GAA (specifying Glu) and GGT (specifying Gly), respectively. There are two nucleotide differences in those codons which presents us with a problem of ordering. Assuming that the nucleotide changes were independent of one another, which of the following paths is correct:

$$GAA(Glu) \Rightarrow GGA(Asp) \Rightarrow GGT(Gly)$$

or

$$GAA(Glu) \Rightarrow GAT(Gly) \Rightarrow GGT(Gly).$$

The first path of substitutions contains two nonsynonymous changes while the second path contains one synonymous and one nonsynonymous change. The path we choose will affect our tabulation of sites and could cause our estimation to be inaccurate.

There is no simple method to handle this problem, however, most methods involve determining the frequency of each codon change. Then, when multiple substitutions have occurred, a cumulative frequency is calculated by multiplying the frequency of each individual change along the path of substitutions. These cumulative frequencies are then used as weights when counting a transition (P_i) or transversion (Q_i). Some have argued that this technique is *ad hoc* and treats the data arbitrarily [81]. Indeed, Li's method uses parsimony tree building (described in section 3) to determine the initial frequencies of all the codons and then combines it with a physiochemical measure of amino acid similarity to group the frequencies from 400 (20 amino acids changing into 20 other amino acids) to four categories. This method would encounter many of the same problems associated with determining the proper distribution of the data (discussed in section 2.3.2) and should only be considered approximate.

4.3 Applying Maximum Likelihood Methods

An alternative method to estimate the rates of synonymous and nonsynonymous substitution is to use an explicit model of codon evolution and to fit the parameters of the model to a set of data using maximum likelihood estimation. There are many advantages to this approach one of which is that knowledge of the substitution process, such as transversion/transition rate bias or codon frequency bias can be incorporated directly into the model. Importantly, the nonsynonymous/synonymous rate ratio (ω) can be allowed to vary over

different parts of the genes being studied and more multiple sequences can be compared at one time (where as Li's approximate method can only be applied to pairs of genes). In fact, all of the models discussed in section 3.3.1 could be used to infer synonymous and nonsynonymous substitution rates. Also, the evolutionary model explicitly describes all the possible paths of substitution for various codons and the maximum likelihood estimation naturally incorporates weighting those paths. The only major disadvantage to the method is the computational burden involved in maximum likelihood estimation – the approximate methods are very fast in comparison [82, 81, 10].

Ziheng Yang applied the following algorithm in [82]. He starts with an evolutionary model similar to Goldman and Yang's codon substituion model described in section 3.3.1. In that model the instantaneous substitution rate from codon i to j is given by

$$Q_{ij} = \begin{cases} 0, & \text{if } i \text{ and } j \text{ differ at more than one position,} \\ \mu\pi_j, & \text{for synonymous transition,} \\ \mu\kappa\pi_j, & \text{for synonymous transversion,} \\ \mu\omega\pi_j, & \text{for nonsynonymous transition,} \\ \mu\omega\kappa\pi_j, & \text{for nonsynonymous transversion.} \end{cases}$$

Parameter κ represents the transition/transversion rate bias, ω represents the nonsynonymous/synonymous rate ratio (K_A/K_S), and π represents the codon background frequency. μ is a scaling factor that is set such that in an instantaneous amount of time the average number of codon substitutions will be one. At this point we use maximum likelihood to estimate the parameters listed above.

Once we have our estimated parameters the proportion of synonymous substitutions is given by

$$\rho_S^* = \sum_{i \neq j} \pi_i Q_{ij}$$

and the proportion of nonsynonymous substitutions is simply $\rho_N^* = 1 - \rho_S^*$. This summation is then calculated for every pair of codons. The number of synonymous and nonsynonymous substitutions per codon is then given by $t\rho_S^*$ and $t\rho_N^*$, respectively where t represents the branch length. These two numbers give us the observed number of substitutions according to our model. We also need to know the expected number of substitutions, or, the number of substitutions that occurred before natural selection had a chance to purify any of them from the organism. These two numbers, ρ_S^1 and ρ_N^1 are calculated in the same way as ρ_S^* and ρ_N^*, except with ω set to one in the model (representing nuetrality with respect to selection). Finally, since there are three nucleotides per codon, the number of nonsynonymous and synonymous substitutions per site are given by

$$K_S = \frac{3t\rho_S^*}{3\rho_S^1} \text{ and}$$

$$K_N = \frac{3t\rho_N^*}{3\rho_N^1}.$$

4.4 Case Study: Measuring the Evolutionary Rates of Duplicate Genes

In [10], John Conery and Michael Lynch sought to discover the fates of genes that were duplicated in large duplication events, such as chromosome rearrangements and whole-genome

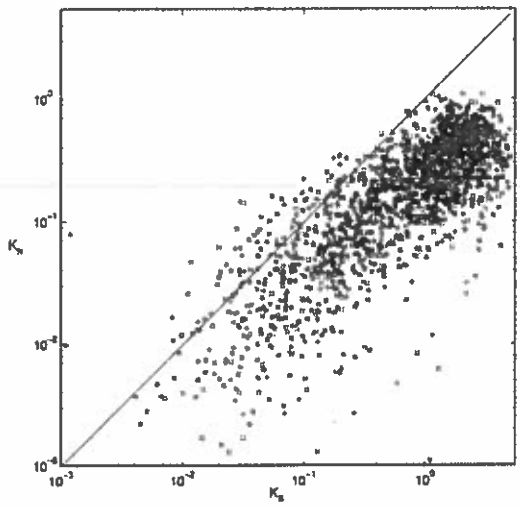


Figure 27: A log-log plot of the nonsynonymous/synonymous substitution rate ratio for nematodes from [10]. The diagonal line, $K_N = K_S$ represents neutral evolution. Gene pairs that fall above the line are experiencing positive selection while pairs below the line are under purifying selection.

duplications (explained in more detail in the section 5). They examined gene duplicates (excluding large gene families) in nine different species including brewer's yeast, nematode, thale cress, humans, and mouse among others. Gene duplicates within organisms were identified using BLAST and gene pairs of interest were filtered using Li's approximate synonymous/nonsynonymous algorithm. Finally, gene pairs of interest were examined using Goldman and Yang's maximum likelihood model. While the full results (and a discussion of what the results imply about gene duplication) are beyond the scope of this document, their analysis produced excellent results that illustrate the concepts of synonymous and nonsynonymous rates of evolution well. Figure 27 shows a log-log K_N/K_S ratio plot of one such result for nematodes. The diagonal line, $K_N = K_S$ represents neutral evolution. Gene pairs that fall above the line are experiencing positive selection while pairs that fall below the line are under purifying selection.

4.5 Conclusions

Algorithms developed to measure nonsynonymous/synonymous rates have been stable since the late 1990's with the publication of Yang's method of using maximum likelihood with a codon-based model. Additionally, the approximate methods, such as the algorithm developed by Li are also still widely used, especially as a preliminary algorithm to identify genes that should be further examined with the more computationally intense methods. There has also been some preliminary work to apply Bayesian inference algorithms to the study of nonsynonymous/synonymous mutation rates such as that by Huelsenbeck in [34]. In terms of the appilcation of the algorithms, they are being widely used to search for positive selection in a variety of genomes, such as in the work of Yang on the human immunodeficiency virus [76].

5 Genome Duplication

While the majority of this position paper has focused on the mechanics of algorithms that define bioinformatics analysis, in this final section we will review the biological evidence for genome duplication. As we stated in the introduction, much of our ability to make inferences about evolutionary events relies on our ability to detect the signal of gene and genome duplication. However, the means and frequency of gene and genome duplications are not an accepted fact. Individual genes are known to be periodically duplicated and, in fact, developmental processes are often regulated by groups or families of genes. For example, the *HOX* genes are known to control the layout of the body plan in an organism's early development. These genes are well conserved across all animals, although the exact number of genes in the families varies – non-vertebrates have a single cluster of *HOX* genes while mammals have four clusters. The primary question we wish to answer is how do these genes duplicate? Are all duplicated genes produced randomly, one at a time with their individual fates decided by natural selection (tandem duplications)? Or, do whole chromosomes, indeed whole genomes duplicate in a single event? In the remainder of this section we will examine the evidence that exists for whole genome duplication events. This evidence has accumulated alongside our ability to measure and analyze increasingly larger amounts of genetic data and we will discuss the evidence historically as it appeared.

Over thirty-five years ago Susumu Ohno postulated that some type of gene duplication was necessary for the emergence of complex organisms [59]. Based on the evidence of the time, which included rough genome size and a count of the number of chromosomes in different organisms, Ohno compared what he thought our earliest ancestors would have been like to similar organisms today. Vertebrates emerged from a simple form 500 million years ago in the Cambrian period. Ohno postulated that these early creatures were similar to tunicates that are alive today, such as *Ciona intestinalis* – a sea squirt. *Ciona* has a genome 6% the size of mammals, while *Amphioxus*, a chordate living today that is most similar to the vertebrates, have genomes approximately 17% the size of mammals. Based on these numbers, Ohno hypothesized that the evolution from tunicate-like creatures to amphioxus-like was accompanied by a two or three-fold increase in genome size. Further, while tandem gene duplications are understood and generally accepted, Ohno argued that they are not enough to create vertebrate complexity. As evidence, he cited that while ancient, primitive fish, which were believed to have undergone only tandem duplication, were able to create large genomes, modern, highly specialized fish, which are descended from these primitive fish, have small, compact genomes – the implication being that large numbers of tandem duplications would create unorganized results, while the process of a whole genome duplication could regulate the eventual number and placement of genes. According to Ohno, the lungfish illustrates this phenomenon very well. With a genome thirty-five times larger than that of mammals, it represents an extreme case where tandem duplication went out of control. The scope of the duplications were so large that the lungfish had to increase the physical size of its cells to accommodate all the genetic material, and this size increase further accelerated the rate of tandem gene duplications in order to make enough gene products to service the larger cells [59].

While this early theory was quite powerful, many of its predictions have turned out not to be correct. The number of chromosomes and the physical size of the genome turns out not to be correlated with gene duplications. All vertebrates, including humans, are believed to have undergone two whole genome duplications and their DNA includes long stretches of non-coding DNA. The pufferfish, on the other hand, may have undergone three genome

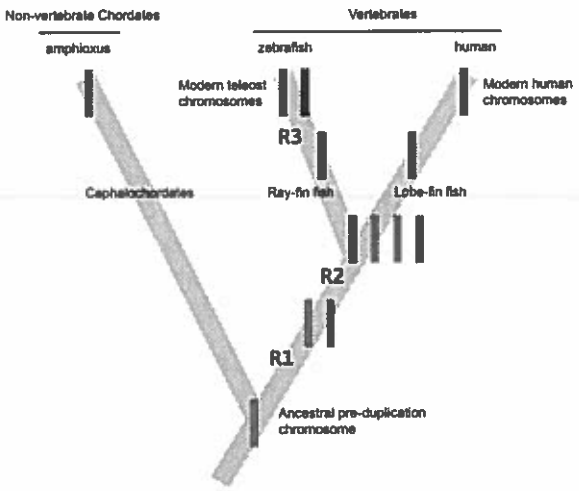


Figure 28: An illustration of the major genome duplication events as proposed in the literature. At the base of the tree lies a single chromosome from an ancient ancestral organism. Branching to the right are the Cephalochordates (a modern example of which is amphioxus) which are believed not to have undergone any genome duplications. To the left the line that leads to modern vertebrates extends, showing the chromosome duplicated after each of the **R1** and **R2** genome duplications. We see the lineage diverge, representing the ray-finned and lobe-finned speciation event, with the teleost fish extending to the right, and the lineage leading to humans on the left. There is believed to have been a third genome duplication (**R3**) in the teleost fish, after splitting from the line leading to humans.

duplications but its genome is very compact. The mechanisms that cause an organism's genome to be more or less compact are not well understood.

5.1 Vertebrate Genome Duplication

In general, proponents of whole genome duplication believe that there has been two full duplications in the history of the lineage leading to humans and a third duplication occurring in the teleost fish after ray-finned and lobe-finned fishes diverged (the former branch led to the teleost fish and the later branch led to humans). These duplications are commonly referred to as "R1", "R2", and "R3" respectively and are believed to have fueled great speciation events that are seen in the fossil record – R1 and R2 at the base of the vertebrate radiation during the Cambrian explosion and R3 at the base of the teleost radiation (teleost fish comprise one of the most diverse collections of species on the planet and include the zebrafish and pufferfish among many others). While there is evidence for other duplications (a 4R event in *Xenopus tropicalis* – a frog, as well as three unrelated genome duplications in the single-celled paramecium), in this paper we will focus on duplications in the lineages leading to humans and the teleost fish.

Figure 28 illustrates the process of duplication and shows the progression of duplicated chromosomes following the R1, R2, and R3 events. What this illustration bears out is that after a duplication, there are two copies of every gene. Over time, positive and negative selection will act on these genes preserving some, creating new function in others (or

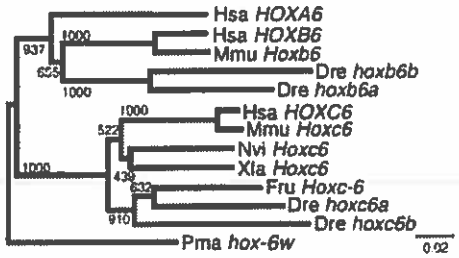


Figure 29: From [6], an image depicting the phylogenetic relationship between zebrafish (Dre), human (Hsa), and mouse (Mmu) *HOX* clusters. This particular tree shows two examples of the duplication topology (a two-to-one ratio of teleost fish genes to mammalian genes. At the top of the image we can see the human *HOXB6* gene most closely related to the mouse *Hoxb6* gene, with the next most closely-related branch in the lineage showing two copies of the orthologous zebrafish *hoxb6a* and *hoxb6b* genes. A similar topology is shown at the bottom of the image with the human *HOXC6* gene.

partitioning function between them), and allowing a large number to become pseudogenes through deleterious mutations. However, for those genes that are conserved we should be able to detect them in a two-to-one ratio relative to the most recent duplication. So, if we compare human genes (two duplications) against *Ciona* (no duplications) we should find a ratio of four-to-one for each duplicated gene; this is often referred to as the *four-to-one* rule. Similarly, some authors have presented the *eight-to-four-to-one* rule to describe the presence of zebrafish (and other teleost fish) genes.

5.2 Examination of the *HOX* Gene Families

While Ohno's analysis was not based on any actual genetic data, in the later half of the 1990's enough protein sequence data became available to begin testing the 3R theory in teleost fish. Amores and colleagues, in [6], examined the *HOX* gene clusters in *Danio rerio*, the zebrafish. As we mentioned above, the *HOX* clusters are responsible for the development of the body plan and are present in organisms from invertebrates to tetrapods (vertebrates having four limbs). While invertebrate chordates only have a single cluster, and little body shape diversity, tetrapods have four clusters, *HOXA* through *HOXD*, and substantial diversity. Amores and colleagues sequenced the *HOX* gene coding regions, examined the chromosomes containing them and analyzed phylogenetic trees of the *HOX* genes with respect to humans. They found that for three of the four mammalian *HOX* clusters, there were two orthologous zebrafish clusters and in the fourth case, a single cluster was found in the zebrafish. Analysis of the genes demonstrated that the zebrafish had two copies of the mammalian *HOXB6* gene, *hoxb6a* and *hoxb6b*, as well as two copies of the mammalian *HOXC6* gene, known as *hoxc6a* and *hoxc6b*; duplicates of the *HOXA* cluster were also found.

The locations of the duplicate zebrafish genes indicate that all four *HOX* clusters were duplicated in the teleost fish, however, one copy of the *HOXD* cluster appeared to have been lost, but is was recently determined that the cluster was reduced to a single gene [75]. The phylogenetic tree showing genes from the *HOXB* and *HOXC* clusters is shown in figure 29 and illustrates the 'duplication topology.' For example, we see that the zebrafish *hoxb6a* and *hoxb6b* genes are more closely related to each other than any other organism, and that human *HOXB6* and the zebrafish genes are more closely related to each other than to the

human *HOXA6* gene. The mouse *Hoxb6* gene confirms the relationship of a single copy of *HOXB6* gene in mammals and two copies in fish. A similar relationship is shown with the zebrafish *hoxc6a* and *hoxc6b* duplicates.

In [71], Taylor performed a computational analysis of existing zebrafish sequences to evaluate the hypothesis that extra *HOX* clusters in zebrafish were produced via a whole-genome duplication event. Taylor downloaded protein sequence data for zebrafish, mouse, chicken, and clawed frog and used BLAST to identify potential orthologs among the various species aligning related orthologs. He then constructed large phylogenetic trees from the related orthologs and searched the tree for sets of genes showing the duplication topology. Taylor then took sets displaying the desired topology and constructed new phylogenetic trees, using human paralogs as outgroups.

From the analysis Taylor found twenty-seven gene-pairs that exhibited the proper phylogenetic relationship, however, only eighteen of the trees had the duplication topology. Further, maximum likelihood analysis turned up only three trees with the proper topology. With regard to the *HOX* genes identified by Amores (in [6]), Taylor only found the topology predicted for *HOXB5* and *HOXB6* genes. By combining their predictions with mapping data, the authors determined that duplicated gene placement is not random, with genes falling on linkage groups 1 and 13, 1 and 9, 2 and 7, 17 and 20, and 3 and 12. Conserved chromosomal regions are consistent with a whole genome duplication, but not necessarily with multiple tandem duplications.

Taylor returned with a more comprehensive study two years later in [70] including other fish species such as the pufferfish. Additional sequence data from mouse, chicken, human, and clawed frog were also used as a basis for phylogenetic reconstruction of orthologous genes. Using BLAST, the authors identified 49 sets of genes with one copy in mouse, human, or chicken, one or two copies in frog, and two copies in zebrafish. For twenty-two of these sets, two orthologs were also identified in the pufferfish. In another twenty of these sets, a single orthologous gene was found in pufferfish. The authors aligned sequences using CLUSTALW and then did two rounds of tree construction, after hand-editing the sequences to remove large gaps and other similar anomalies. Phylogenetic trees were constructed using the neighbor-joining method as well as the quartet-puzzling method.

In 24 cases, both phylogenetic methods revealed the proper genome duplication topology. In fourteen cases, one or the two methods revealed the proper topology. Surprisingly, after removing saturated positions in the sequences, the trees built from these sequences found the proper duplication topology in 37 cases, including five cases in which the prior methods were unable to find the sought after topology. In eighteen cases of duplicated genes found in either the zebrafish, the pufferfish, or both, orthologs from other fish species, including salmon and eel, were also identified. Of all the zebrafish orthologs identified, forty-four of them have been mapped, and among these, ten chromosome pairs contain two or more sets of gene duplicates. This number of duplicates is significantly higher than one would expect by chance.

These early studies made fascinating predictions about a third duplication in the teleost fish. Although Taylor's study suffered from limited data availability and computationally underpowered analysis algorithms, Amores' work provided good evidence that entire chromosomes had indeed been duplicated in the zebrafish. Data at the time was not comprehensive enough to avoid criticism, however, as we will see in the next section.

5.2.1 Genome Duplication Criticism

The early studies were very controversial and several people argued that other, simpler mechanisms were more likely to be the cause of the documented duplicated genes. One of the largest critics of genome-wide duplication, in [39] Hughes and Friedman assembled evidence against the 2R hypothesis. Their criticisms challenged some of the essential concepts of duplication, including genome size and gene counts.

Although Ohno argued genome size implied duplication, Hughes argues that the vast number of different genome sizes debunks this hypothesis (bony fish vary from 11%-4088% the size of the human genome). Additionally, while many have cited gene number as evidence of duplication, Hughes claims the four-to-one rule only applies in a small number of genes (4.9%) when comparing human to *Drosophila* (fruit fly) and that among those genes the proper phylogenetic topologies are missing.

Finally, supporters of the 2R theory point to conserved genomic regions on human chromosomes 1, 6, 9, and 19 along with 2, 7, 12, and 17. For these to be correct, according to Hughes, the genes must be phylogenetically dated, and the dates must be the same. Hughes claimed that these duplicated gene families have widely varying duplication dates, but, as we saw in section 3.3.2, it is now well accepted that genes can evolve at different rates, so his criticism must be examined in this light.

In [62], Robinson-Rechavi examined homologous gene sequences from fish gene families that were known to exist in at least three species. For each gene family they then built phylogenetic trees. Their analysis showed that only seven gene families out of the 37 they analysed exhibited characteristics of a genome duplication. Eleven of the gene families showed that duplications had occurred, but that they took place after the teleost radiation. Finally, in the 19 remaining cases, no duplications were observed.

5.3 Whole-Genome Examination

Early studies in favor of duplication, along with early criticisms both suffered from a lack of data rendering the hypothesis for a third duplication in teleost fish controversial. However, that situation began to change as sequence libraries became more comprehensive and entire genomes became available for analysis, bring more evidence to bear in favor of the hypothesis [40].

Vandepoele examined the entire *Takifugu rubripes* (Fugu or pufferfish) genome in search of duplication events [74]. Fugu was initially chosen to be sequenced several years ago due to its extremely compact genome size, which is approximately one-eighth the size of the human genome, however, after *HOX* clusters from other fish implied the possibility of a third genome duplication, the Fugu genome also became a target to search for duplication events.

The authors obtained approximately 8,500 scaffolds (partially assembled chromosomes), containing over 34,000 genes to search for duplications. They then used BLAST to identify orthologous Fugu genes in the human genome and identified gene families from the results. The largest gene families were removed from the set of genes to be analyzed due to the difficulty of constructing phylogenetic trees out of them (due to problems of long-branch attraction). Next, they used proteins from these gene families as BLAST queries for other species including all other ray-finned fish, humans, mice, *Ciona*, and *Drosophila* to identify an outgroup, aligned the results and built phylogenetic trees. To associate dates to the various duplications, Vandepoele performed an analysis using a molecular clock.

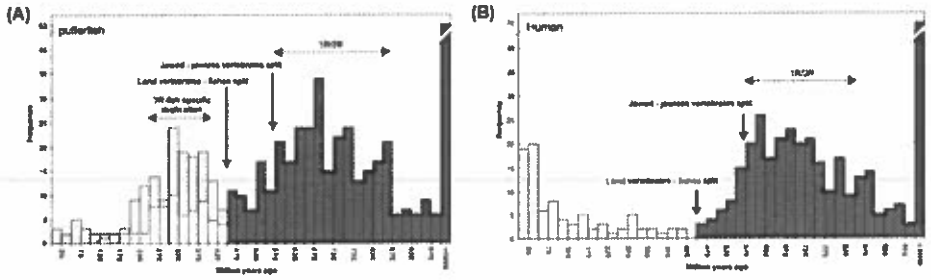


Figure 30: From [74], (A and B) the age distribution of duplicated genes identified in Fugu (pufferfish) and human respectively. The 3R event only appears in the pufferfish lineage, while human duplications appear to have accelerated in recent time. Additionally, the white bars in (A) refer to pufferfish genes that were duplicated on chromosomal blocks, the location of these genes provides evidence for large scale duplication events, such as a whole genome duplication.

The resulting phylogenetic trees, with branch lengths calibrated using a molecular clock, produced 166 genes (30%) originating in the 3R duplication and 399 genes originating in the 1R or 2R duplications. A similar analysis was conducted using the human genome in order to compare the results to Fugu. Figure 30 shows the distribution of gene ages identified in the analysis of Fugu and human genes. The major feature the plot illustrates is that the 3R duplication in Fugu does not correspond to an increase in gene duplication in humans, which we would expect if only one of the organisms experienced the duplication. Further, if the plot was only showing fluctuations in tandem duplications we would not expect to see such a stark contrast between the two genomes. To further investigate the origin of the 3R Fugu genes Vandepoele investigated whether the duplicated genes appeared together in conserved, duplicated blocks (also known as conserved synteny) since successive tandem duplications would not preserve gene ordering on chromosome segments. Vandepoele mapped the paralogous genes to their respective scaffolds and searched for conserved gene ordering (conserved synteny) and gene age. The authors were able to identify 159 statistically significant duplicated blocks among the scaffolds, containing 544 gene pairs. 59 3R-dated blocks were identified and these blocks all demonstrate the same duplication date, approximately 320 million years ago.

At the same time, Christoffels also performed a genome-wide analysis using early drafts of the Fugu genome in [8]. In a very similar analysis, Christoffels identified gene families, constructed phylogenetic trees using a molecular clock and searched for conserved regions among the scaffolds. Their results agreed quite well with Vandepoele producing 468 Fugu paralogons covering 6.8% of the genome as well as 425 duplicate gene pairs. Their clock analysis found that the R3 duplication occurred approximately 350 million years ago.

Jaillon and colleagues describe the *Tetraodon nigroviridis* (pufferfish) genome sequence in [41]. This fish, which is related to Fugu, but lives in freshwater, has a similarly compact genome and the two organisms diverged only 18 to 30 million years ago. The release of this full genome consisted of 340Mb of genetic material composing 27,918 genes on 21 chromosomes. While Jaillon and co-workers provided a wealth of data regarding the pufferfish genome we will concentrate on their evidence for whole genome duplication here.

Jaillon sought evidence for two distinctive signs of a whole-genome duplication within the

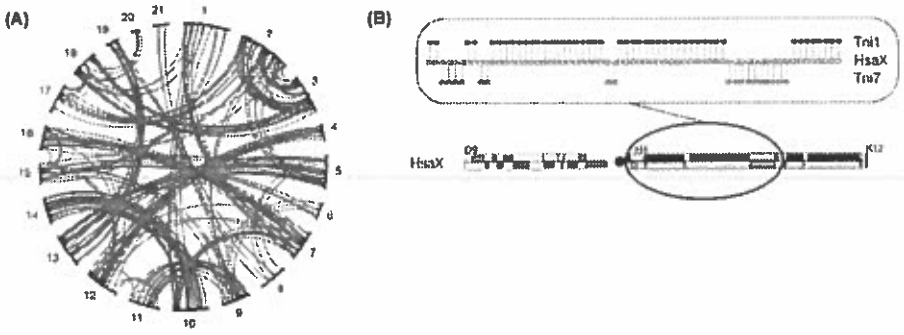


Figure 31: From [41], (A) an image depicting paralogous *Tetraodon nigroviridis* genes and the chromosomes they fall on. Some of the duplicated chromosomes have remained mostly intact since the duplication (e.g. 9 and 11) while others have been rearranged onto other chromosomes (e.g. 13 being split between 19 and 5). (B) Conserved syntenic regions between human chromosome X and paralogous regions located on pufferfish chromosomes 1 and 7. The interleaving pattern of genes is indicative of a genome duplication followed by massive gene loss.

pufferfish: paralogous gene pairs within the pufferfish that are arranged on separate chromosomes, and, the characteristic two-to-one ratio of orthologous genes when the pufferfish was compared against a genome that had not experienced the duplication. To search for evidence of paralogous gene pairs Jaillon and colleagues performed a Tetraodon to Tetraodon BLAST search, looking for reciprocal best hits. This analysis yielded 1,078 pairs of duplicated genes within the pufferfish and is shown in figure 31 (A). As the plot illustrates, some chromosomes have experienced almost no rearrangements since the duplication event, such as pufferfish chromosomes 9 and 11, while others have split apart or been fused together onto other chromosomes (genes paralogous to chromosome 13 are found on chromosome 19 and 5, for example).

To search for evidence of a two-to-one ratio, Jaillon compared 6,684 Tetraodon genes on known chromosomes to their mouse and human orthologs. In the case of a genome duplication one would expect to find chromosomal segments in human or mouse that correspond to two paralogous chromosomal segments in pufferfish (these segments being identified by the orthologous genes that are contained within them). Jaillon created a syntenic map from the sets of orthologous genes containing 900 syntenic groups where each group was composed of at least two consecutive, ordered genes. The average length of these regions was 6.1 genes, with a maximum of 55 genes. Of the 6,684 genes, 76% ended up in a syntenic group. Similar results were found in a comparison with mouse. One example of the syntenic mapping can be see in figure 31 (B) which shows a large segment of human chromosome X corresponding to paralogous regions in pufferfish chromosomes 1 and 7. The genes in these regions are interleaved between the two pufferfish chromosomes indicating a whole genome duplication followed by massive gene loss. Jaillon asserts that these two analyses provide definitive evidence in favor of a whole genome duplication in the pufferfish after it split from the land vertebrates.

Finally, Dehal and Boore performed an analysis of three vertebrate species along with *Ciona intestinalis* in search of evidence of the 2R duplication [12]. Their analysis started by searching for orthologous genes between human, mouse, Fugu, and *Ciona*. In the same

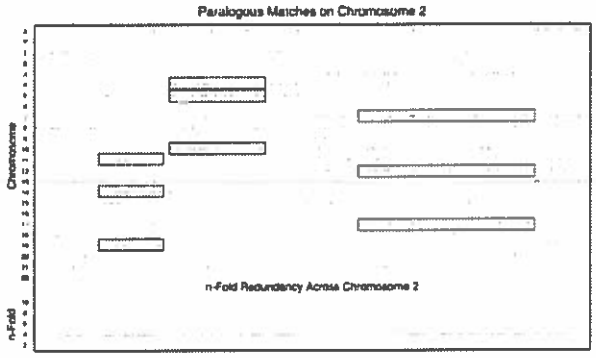


Figure 32: From [12], evidence for the 2R duplication in vertebrates. This plot shows human genes paralogous to genes on human chromosome 2. The boxes highlight several regions where the 2R signal is evident (three conserved regions along with the region on chromosome 2 which is not pictured). Along the bottom of the plot are the results of a sliding window analysis showing how many conserved segments exist and the chromosomes they occur on.

way that Vandepoele, Christoffels, and Jaillon all sought evidence of a two-to-one ratio in teleost fish to human genes, Dehal and Boore were looking for a four-to-one ratio between the vertebrates (human, mouse, Fugu) and the invertebrate *Ciona* (which is not believed to have undergone a genome duplication). To search for these relationships, after identifying orthologs between all the organisms, the authors determined which of the vertebrates that had the highest raw BLAST score for a particular *Ciona* gene. They then used a clustering algorithm to find all vertebrate genes (clustering on raw BLAST score) that had a higher similarity to that vertebrate gene than to any other *Ciona* gene. From the clusters they built phylogenetic trees using a maximum likelihood algorithm. They examined the resulting trees for a duplication topology as well as to identify paralog genes within different species. They plotted the paralog genes and finally used a sliding window analysis to find paralogons that show the right topology (the 2R signal would be found as conserved regions on four different chromosomes within an organism). An example of this analysis is shown in figure 32. Pictured are human genes identified as paralogs to genes found on human chromosome two. Along the bottom of the graph is the results of the sliding window analysis showing the conserved segments of chromosomes. Dehal and Boore's approach of building related gene clusters and then analyzing the topology of the resulting phylogenetic trees provides startling evidence for the 2R duplication in the form of the paralog graphs; although the duplication occurred 450 million years ago, the signal of the event is easily discernable.

It has been almost ten years since Amores published his work proposing a third genome duplication in zebrafish. Since that time, with the work of Jaillon, and Dehal and Boore, among others, evidence for genome duplication events has become widespread. The signal for these events is now well understood and several robust methods exist to identify it. In the next and final section, we will summarize the major efforts of this paper and look at what open questions remain to be investigated.

6 Conclusions

This paper has focused on the theory and methods necessary to investigate the hypothesis of whole genome duplication: a process in which a single, rare event doubles an organism's chromosomes providing the raw, genetic material necessary to spawn a multitude of species. In section 2, we examined the algorithms and statistics involved in sequence alignment, a process that allows us to determine relationships between genes and to identify conserved regions of DNA within organisms as well as between organisms. In section 3, we discussed the process of building phylogenetic trees, along with the statistical models on which they depend, allowing us to quantify the evolutionary history of features in a set of related genes. In section 4, we looked at the neutral theory of evolution and showed how it could be used to infer the strength of natural selection through the measurement of nonsynonymous/synonymous mutation rates. Finally, in section 5 we looked at the empirical evidence for the existence of genome duplication events.

As section 5 demonstrated, strong evidence now exists in favor of whole genome duplications. The signal of genome duplications has been found in many different species, from the teleost fish to the single-celled paramecium. Several challenges remain, however, to more fully understand the processes that occur after the duplication is complete. Specifically, we want to know what are the fates of duplicated genes? The theory of neofunctionalization states that after a duplication one of the genes retains the old function while the other is free to acquire new functions. This theory predicts a chaotic environment in an organism's cells following a duplication: the proportion of transcripts produced in the cell would all increase as the doubled genes begin producing double the number of proteins! New genes would be large targets for mutations, causing a lot of genes to experience death shortly after their creation as terminal mutations cause natural selection to purge the gene.

In contrast, the theory of subfunctionalization predicts a more stable environment following duplication. In this scenario, gene functions become partitioned between the original copy of the gene and the new copy of the gene making both copies indispensable. This strategy would retain the proper proportions of gene transcripts and provide the copied genes time to acquire new, useful functions as natural selection would continue to preserve both copies in the population [28].

Continuing work in the area of whole genome duplication needs to focus on the fates of duplicated genes, which can be done in three ways. First, more work needs to be done to quantify the number and location of duplicated genes among different organisms. We must continue to identify all the duplicates in a range of organisms from teleost fish, to mammals, insects, and non-vertebrates. Moreover, we must quantify orthologous regions between these species to determine which genes were conserved in which species. This work would enable the creation of a duplication map. A researcher would query the map with a particular gene and the map would return the duplicated region the gene is located in (if one exists) as well as providing all orthologous duplicated regions from other species. By comparing and contrasting genes that have been preserved and lost in different species, we may be able to infer the general behavior of gene loss following a duplication. Additionally, these maps will allow us to infer the location of specific, lost genes. For example, if we identify a duplicated gene in one species, but cannot find it in another, we may be able to infer its absence by looking at the surrounding genes in the duplicated region. Using a large scale analysis of multiple organisms, how many lost genes can we find?

A second question we can address with our duplication map is the behavior of natural selection following the duplication. Is natural selection active in genes that have been dupli-

cated? Is it only active in one copy of the gene, or is it active in subregions of both copies? Further, is natural selection active in different subregions of the same genes in different species? Answering this question does pose some challenges, primarily that some duplications happened too far in the past to apply our measures of nonsynonymous/synonymous mutation rates (due to saturation at some sites in the codons we would like to measure). An effective analysis of the strength of natural selection will require testing more recently duplicated genes.

Finally, we can examine the hypothesis of subfunctionalization itself. As mentioned in section 1.1, genes contain regulatory regions upstream from their coding regions. These regulatory regions control when a gene is transcribed and the rate at which it is transcribed. If a gene has multiple functions, there are generally multiple regulatory regions upstream of the gene to control those functions. After a genome duplication event, if gene functions are indeed segregated between the two copies, we would expect to see the loss of different regulatory regions in each of the two duplicate genes. We can investigate this hypothesis by examining the regions of nongenic DNA upstream of our pairs of duplicated genes by looking for conserved sequence data in those areas. If we find significant but differing conservation in each gene copy, then it would indicate that the regulatory regions were indeed partitioned.

The tools we have described in this paper will allow us to complete these analyses. Through the coordinated application of these tools, using substantial computational resources to execute, analyze, and store the resulting data we can answer questions about the fate of duplicated genes furthering the state of the art in the bioinformatics of whole genome duplication.

References

[1] S. F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219(3):555–565, June 1991.

[2] S. F. Altschul, M. S. Boguski, W. Gish, and J. C. Wootton. Issues in searching molecular sequence databases. *Nature Genetics*, 6(2):119–129, February 1994.

[3] S. F. Altschul and W. Gish. Local alignment statistics. *Methods in Enzymology*, 266:460–480, 1996.

[4] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, May 1990.

[5] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI–BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[6] A. Amores, A. Force, Y.-L. Yan, L. Joly, C. Amemiya, A. Fritz, R. K. Ho, J. Langeland, V. Prince, Y.-L. Wang, M. Westerfield, M. Ekker, and J. H. Postlethwait. Zebrafish hox clusters and vertebrate genome evolution. *Science*, 282(5394):1711 – 1714, November 1998.

[7] H. Carrillo and D. Lipman. The multiple sequence alignment problem in biology. *Journal of Applied Mathematics in Biology*, 48(5):1073–1082, October 1988.

[8] A. Christoffels, E. G. L. Koh, J. ming Chia, S. Brenner, and S. Aparici. Fugu genome analysis provides evidence for a whole-genome duplication early during the evolution of ray-finned fishes. *Molecular Biology and Evolution*, 21(6):1146–1151, 2004.

[9] F. S. Collins, M. Morgan, and A. Patrinos. The human genome project: Lessons from large-scale biology. *Science*, 300(5617):286 – 290, April 2003.

[10] J. Conery and M. Lynch. Nucleotide substitutions and the evolution of duplicate genes. *Pacific Symposium on Biocomputing*, 6:167–178, 2001.

[11] M. Dayhoff, R. Schwartz, and B. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352, 1978.

[12] P. Dehal and J. L. Boore. Two rounds of whole genome duplication in the ancestral vertebrate. *PLoS Biology*, 3(10):1700–1708, October 2005.

[13] C. Do, M. Mahabhashyam, M. Brudno, and S. Batzoglou. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15:330–340, 2005.

[14] R. F. Doolittle. Similar amino acid sequences: chance or common ancestry? *Science*, 214(4517):149–159, October 1981.

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

[16] S. Eddy. What is a hidden markov model? *Nature Biotechnology*, 22:1315–1316, 2004.

[17] S. Eddy. What is dynamic programming? *Nature Biotechnology*, 22:909–910, 2004.

[18] R. Edgar. Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Research*, 32(1):380–385, 2004.

[19] R. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[20] B. Efron, E. Halloran, and S. Holmes. Bootstrap confidence levels for phylogenetic trees. *Proceedings of the National Academy of Sciences of the USA*, 93:13429–13429, November 1996.

[21] J. Felsenstein. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology*, 27(4):401–410, December 1978.

[22] J. Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, June 1981.

[23] J. Felsenstein. Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, 39(4):783–791, July 1985.

[24] J. Felsenstein. Phylogenies from molecular sequences: inferences and reliability. *Annual Review Genetics*, 22:521–565, 1988.

[25] D.-F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4):351–360, 1987.

[26] W. M. Fitch. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20(4):406–416, December 1971.

[27] W. M. Fitch and T. F. Smith. Optimal sequence alignments. *Proceedings of the National Academy of Sciences of the USA*, 80(5):1382–1386, March 1983.

[28] A. Force, M. Lynch, F. B. Pickett, A. Amores, Y. lin Yan, , and J. Postlethwait. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151:1531–1545, 1999.

[29] P. G. Foster. The idiot's guide to the zen of likelihood in a nutshell in seven days for dummies, unleashed. 2001.

[30] N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding dna sequences. *Molecular Biology and Evolution*, 5:725–736, September 1994.

[31] G. H. Gonnet, M. A. Cohen, and S. A. Benner. Exhaustive matching of the entire protein sequence database. *Science*, 256(5062):1443–1445, June 1992.

[32] S. Henikoff and J. Henikoff. Automated assemble of protein blocks for database searching. *Nucleic Acids Research*, 19(23):6565–6572, November 1991.

[33] S. Henikoff and J. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the USA*, 89:10915–10919, 1992.

[34] J. Huelsenbeck and K. Dyer. Bayesian estimation of positively selected sites. *Journal of Molecular Evolution*, 58(6):661–672, June 2004.

[35] J. P. Huelsenbeck and K. A. Crandall. Phylogeny estimation and hypothesis testing using maximum likelihood. *Annual Review Ecological Systems*, 28:437–466, 1997.

[36] J. P. Huelsenbeck, B. Larget, R. E. Miller, and F. Ronquist. Potential applications and pitfalls of bayesian inference of phylogeny. *Systematic Biology*, 51(5):673–688, October 2002.

[37] J. P. Huelsenbeck and F. Ronquist. Mrbayes: Baysian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, March 2001.

[38] J. P. Huelsenbeck, F. Ronquist, R. Nielsen, and J. P. Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294:2310–2314, December 2001.

[39] A. L. Hughes and R. Friedman. 2r or not 2r: testing hypotheses of genome duplication in early vertebrates. *Journal of Structural and Functional Genomics*, 3:85–93, 2003.

[40] N. Hukriede, D. Fisher, J. Epstein, L. Joly, P. Tellis, Y. Zhou, B. Barbazuk, K. Cox, L. Fenton-Noriega, C. Hersey, J. Miles, X. Sheng, A. Song, R. Waterman, S. L. Johnson, I. B. Dawid, M. Chevrette, L. I. Zon, J. McPherson, and M. Ekker. The ln54 radiation hybrid map of zebrafish expressed sequences. *Genome Research*, 11:2127–2132, 2001.

[41] O. Jaillon, J.-M. Aury, F. Brunet, J.-L. Petit, N. Stange-Thomann, E. Mauceli, L. Bouneau, C. Fischer, C. Ozouf-Costaz, A. Bernot, S. Nicaud, D. Jaffe, S. Fisher, G. Lutfalla, C. Dossat, B. Segurens, C. Dasilva, M. Salanoubat, M. Levy, N. Boudet,

S. Castellano, V. Anthouard, C. Jubin, V. Castelli, M. Katinka, B. Vacherie, C. Biémont, Z. Skalli, L. Cattolico, J. Poulain, V. de Berardinis, C. Cruaud, S. Duprat, P. Brottier, J.-P. Coutanceau, J. Gouzy, G. Parra, G. Lardier, C. Chapple, K. J. McKernan, P. McEwan, S. Bosak, M. Kellis, J.-N. Volff, R. Guigó, M. C. Zody, J. Mesirov, K. Lindblad-Toh, B. Birren, C. Nusbaum, D. Kahn, M. Robinson-Rechavi, V. Laudet, V. Schachter, F. Quétier, W. Saurin, C. Scarpelli, P. Wincker, E. S. Lander, J. Weissenbach, and H. R. Crollius. Genome duplication in the teleost fish tetraodon nigroviridis reveals the early vertebrate proto-karyotype. *Nature*, 431(7011):946–957, October 2004.

[42] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the USA*, 87:2264–2268, March 1990.

[43] M. Kimura. A simple method for estimating evolutionary rates of base substitution through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, June 1980.

[44] M. Kimura and T. Ohta. On some principles governing molecular evolution. *Proceedings of the National Academy of Sciences of the USA*, 71(7):2848–2852, May 1974.

[45] J. King and T. Jukes. Non-darwinian evolution. *Science*, 164(3881):788–798, May 1969.

[46] B. Kolaczkowski and J. W. Thornton. Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous. *Nature*, 431(21):980–984, October 2004.

[47] Z. Kozmik. Pax genes in eye development and evolution. *Current Opinion in Genetics and Development*, 15(4):430–438, August 2005.

[48] J. B. Kruskal. An overview of sequence comparison: time warps, string edits, and macromolecules. *SIAM Review*, 25(2):201–237, April 1983.

[49] D. Leja. Gene, National Human Genome Research Institute (http://www.accessexcellence.org/rc/vl/gg/gene.html).

[50] D. Leja. mrna, National Human Genome Research Institute (http://www.accessexcellence.org/RC/VL/GG/mRNA.html).

[51] W.-H. Li. *Molecular Evolution*. Sinauer Associates, 1997.

[52] W.-H. Li, C.-I. Wu, and C.-C. Luo. A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes. *Molecular Biology and Evolution*, 2(2):150–174, March 1985.

[53] D. Lipman, S. Altschul, and J. Kececioglu. A tool for multiple sequence alignment. *Proceedings of the National Academy of Sciences of the USA Biochemistry*, 86:4412–4415, June 1989.

[54] D. J. Lipman, W. J. Wilbur, T. F. Smith, and M. S. Waterman. On the statistical significance of nucleic acid similarities. *Nucleic Acids Research*, 11(12):215–226, January 1984.

[55] A. Löytynoja and N. Goldman. An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the USA*, 102(30):10557–10562, July 2005.

[56] M. Margulies, M. Egholm, W. E. Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y.-J. Chen, Z. Chen, S. B. Dewell, L. Du, J. M. Fierro, X. V. Gomes, B. C. Godwin, W. He, S. Helgesen, C. H. Ho, G. P. Irzyk, S. C. Jando, M. L. I. Alenquer, T. P. Jarvie, K. B. Jirage, J.-B. Kim, J. R. Knight, J. R. Lanza, J. H. Leamon, S. M. Lefkowitz, M. Lei, J. Li, K. L. Lohman, H. Lu, V. B. Makhijani, K. E. McDade, M. P. McKenna, E. W. Myers, E. Nickerson, J. R. Nobile, R. Plant, B. P. Puc, M. T. Ronan, G. T. Roth, G. J. Sarkis, J. F. Simons, J. W. Simpson, M. Srinivasan, K. R. Tartaro, A. Tomasz, K. A. Vogt, G. A. Volkmer, S. H. Wang, Y. Wang, M. P. Weiner, P. Yu, R. F. Begley, and J. M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437:376–380, September 2005.

[57] D. Mount. *Bioinformatics: sequence and genome analysis.* Cold Spring Harbor Laboratory Press, 2001.

[58] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.

[59] S. Ohno. *Evolution by Gene Duplication.* Springer-Verlag, 1970.

[60] T. Ohta. The nearly neutral theory of molecular evolution. *Annual Review Ecological Systems*, 23:263–286, 1992.

[61] C. Orengo, D. Jones, and J. Thornton, editors. *Bioinformatics: genes, proteins and computers.* BIOS Scientific Publishers Limited, 2003.

[62] M. Robinson-Rechavi, O. Marchand, H. Escriva, and V. Laudet. An ancestral whole-genome duplication may not have been responsible for the abundance of duplicated fish genes. *Current Biology*, 11(12):R458–R459, June 2001.

[63] F. Ronquist. Fast fitch-parsimony algorithms for large data sets. *Cladistics*, 14:387–400, November 1998.

[64] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, July 1987.

[65] M. J. Sanderson and J. Kim. Systematic biology. *Systematic Biology*, 49(4):817–829, 2000.

[66] P. H. Sellers. On the theory and computation of evolutionary distances. *SIAM Journal of Applied Math*, 26(4):787–793, June 1974.

[67] H. Smith, T. Annau, and S. Chandrasegaran. Finding sequence motifs in groups of functionally related proteins. *Proceedings of the National Academy of Sciences of the USA Biochemistry*, 87:826–830, January 1990.

[68] T. F. Smith and M. S. Waterman. Comparison of biosequences. *Advances in Applied Mathematics*, 2:482–489, 1981.

[69] D. Swofford, G. Olsen, P. Waddell, and D. Hillis. *Molecular Systematics*, chapter Phylogenetic Inference, pages 407–514. Number 11. Sinauer Associates, 2 edition, 1996.

[70] J. S. Taylor, I. Braasch, T. Frickey, A. Meyer, and Y. V. de Peer. Genome duplication, a trait shared by 22,000 species of ray-finned fish. *Genome Research*, 13:382–390, 2003.

[71] J. S. Taylor, Y. V. de Peer, I. Braasch, and A. Meyer. Comparative genomics provides evidence for an ancient genome duplication event in. *Philosophical Transactions: Biological Sciences*, 356:1661–1679, 2001.

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

[73] J. Thornton. Resurrecting ancient genes: experimental analysis of extinct molecules. *Nature Reviews Genetics*, 5:366–375, 2004.

[74] K. Vandepoele, W. D. Vos, J. S. Taylor, A. Meyer, and Y. V. de Peer. Major events in the genome evolution of vertebrates: Paranome age and size differ considerably between ray-finned fishes and land vertebrates. *Proceedings of the National Academy of Sciences of the USA*, 101(6):1638–1643, February 2004.

[75] J. Woltering and A. Durston. The zebrafish hoxdb cluster has been reduced to a single microrna. *Nature Genetics*, 38:601–602, 2006.

[76] W. Yang, J. Bielawski, and Z. Yang. Widespread adaptive evolution in the human immunodeficiency virus type 1 genome. *Journal of Molecular Evolution*, 57(2):212–221, August 2003.

[77] Z. Yang. Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 39(1):105–111, July 1994.

[78] Z. Yang. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution*, 39(3):306–314, September 1994.

[79] Z. Yang. Among-site rate variation and its impact on phylogenetic analyses. *Trends in Ecology and Evolution*, 11(9):367–372, September 1996.

[80] Z. Yang. *Mathematics of Evolution and Phylogeny*, chapter Bayesian Inference in Molecular Phylogenetics, pages 63–90. Oxford University Press, 2005.

[81] Z. Yang and J. P. Bielawski. Statistical methods for detecting molecular adaptation. *Trends in Ecology and Evolution*, 15(12):496–503, December 2000.

[82] Z. Yang and R. Nielsen. Synonymous and nonsynonymous rate variation in nuclear genes of mammals. *Journal of Molecular Evolution*, 46:409–418, 1998.

[83] E. Zuckerkandl and L. Pauling. Molecules as documents of evolutionary history. *Journal of Theoretical Biology*, 8:357–366, 1965.