A Computer Model of Sequence Mutation, Molecular Distance Measures, and the Parsimony Principle

by

Kevin P Scott

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Pharmaceutical Chemistry

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA

San Francisco

[redacted]

Date **University Librarian**

Degree Conferred: . . . .

Acknowledgements

First, I would like to thank my family. My immediate family for their direction and unwavering support throughout my life. My extended family, particularly the Jeffreys (Mark, Judith and my goddaughter Angela) who have provided not a house but a home for me in San Francisco over the past year. Most of all I would like to thank my wife. Even if I already had my thesis or never wanted a thesis, I would be grateful to have such a wonderful spouse. The fact that she has been so supportive of my returning to graduate school only deepens and broadens my love and respect for her.

Second, I would like to thank UCSF. I came back after a long hiatus. I was shown opportunities by my advisor, by the Pharm Chem department, and by the graduate division. The Dill group, or Dill groups, deserve thanks: I have been a CPU hog among two generations of Dill graduate students. Thanks to my thesis committee for reviewing my old work and my new work. Special thanks go to Patricia Babbitt who took an interest in me and my research, providing references and giving of her own time. Most of all I would like to thank Ken. Beyond the help and review that all advisors give their students (and here I have to say that Ken had to go an extra distance to help me to clarify my writing) I have to respect him for having a humanist approach to people while maintaining an objective approach to science. These two different skills give him an extraordinary ability to produce people of science.

Third, I want to thank my church. I used to see being a scientist and belonging to a church as mutually exclusive. I knew and liked Unitarians and thought well of it, as churches go. As I got to know the church in Wilmington better I was surprised to learn that Charles Darwin was a Unitarian. I am not born again, have not changed my reviews on religion significantly (actually they have grown a little more skeptical), and do not expressly recommend my church or any other to the general population. But I must acknowledge the spiritual grounding it affords me when approaching scientific problems, philosophical problems, or the large spectrum of overlap between them.

A Computer Model of Sequence Mutation, Molecular Distance Measures, and the Parsimony Principle

by

Kevin Paul Scott

Abstract

We devise a simple computer model to study similarities among biomolecule sequences such as DNA or protein molecules. We use a model of exhaustive sequence mutation whereby a given parent sequence undergoes every possible event - substitution, deletion, and insertion - at every mutation site to result in an ensemble of daughter sequences. Those daughters are then subjected to the same process to create and ensemble of second generation daughters, etc. A series of mutations can be described as a "pathway". There are many different pathways that can lead from any parent to any daughter. This model of evolution allows us to explore the concept of "closeness" or "evolutionary relatedness" also referred to as distance or "sequence similarity".

Sequence similarity is often measured by Hamming or Levenshtein distances, which are based on the parsimony principle. Parsimony, as applied to distance measurement, measures similarity as the fewest number of mutations that convert one sequence to another. But our mutational model shows that parsimony sometimes

errs in rank ordering the closeness of sequence relationships. We find that evolutionary distance depends not only the number of mutations used to convert one sequence to another, but also on the composition of characters in a sequence in terms of order. We find that homogeneous sequences (i.e. having all the same character: {a a a ...}) are more interconnected by mutational pathways than heterogeneous sequences (i.e. having different characters: {a b c ...}). We introduce here the notion that the number of pathways between two sequences affects the evolutionary relatedness. We define the *kinetic accessibility* of daughter sequences from their parent sequences to reflect this statistical nature.

We follow the time evolution of mutational processes in this model. Over time, a homogeneous sequence will develop a diversity of characters, and over time a perfectly heterogeneous sequence will grow somewhat "ordered" and a perfectly homogeneous sequence will grow "disordered". In this regard sequence evolution has an arrow of time. Sequence {a a a} is more likely the parent of {a a b} than the reverse. We hope that these simple model studies will be useful for more accurate construction of phylogenetic trees from biomolecular sequences.

Contents

Tables:

Figures:

Chapter 1

Introduction

Distance

This thesis focuses on the measurement of dissimilarity or "distance" between sequences of DNA or amino acids, with emphasis on the parsimony principle. The main use of distance measures is to establish "phylogenies", the graphical trees describing the families and interrelatedness among species. While the preponderance of modern efforts focus on comparing sequences of biomolecules such as protein or DNA, the construction of phylogenic trees predates these efforts. Phylogenies were originally constructed by eyeball comparisons of wings, feather colors, bones, tail length and other anatomical or morphological features. Defining these interrelationships is called cladification. Organisms, or Operational Taxonomic Units (OTUs) are compared in many different features, or categories, simultaneously. The features used in morphology may be quantitative or qualitative, which "may require some ingenuity as well as some arbitrariness in coding states" (Camin and Sokal 1965). What does similarity mean in these cases?

On the other hand molecular data is sequential. A monomer position in a biomolecule sequence is itself a "feature". These features are dynamic: positions may be created or destroyed. But features which change position in a sequence may still be compared to another unchanged sequence position because the possible character states for any position are identical to those at other positions.

Of fundamental importance to molecular sequence comparison, and the classification of phylogenetic trees, is the concept of parsimony. For sequence comparison, parsimony is the idea that the distance between two sequences can be measured in terms of the fewest mutations that convert one sequence to the other. Of course true biological evolution from one sequence A to another sequence B need not have occurred through the smallest number of mutations. A deletion of a character from sequence A can be followed by reinsertion of the same character to leave the sequence unchanged. But how can we know whether, or how often, this happens? Parsimony is not an idea that is based on any real model of evolution or that necessarily reflects the true evolutionary distances between sequences. But parsimony has the advantages of being simple, unambiguous and computationally inexpensive. Two simple distance measures are based on parsimony: Hamming and Levenshtein (see Kruskal 1983). The Hamming distance applies when there are no insertions and deletions and counts the number of mismatches between two aligned sequences. The Levenshtein distance is more general and applies for sequence gaps: insertions and deletions are counted as mismatching positions.

Using simulations of morphological data, Camin and Sokal (1965) performed the first objective test of the principle of parsimony as applied to phylogeny. They found the method to be systematic, it gave trees that agree with known phylogenies, and is suitable for implementation on a computer. Eck and Dayhoff (1966) made use of a similar scheme for DNA sequences. The use of parsimony for DNA sequences is simpler than the use of parsimony for morphological features. Each symbol in DNA has equal weight and these symbols can be compared from one sequence position to the next. But for morphological characteristics and their phenetic states, how do we compare wings to feet? While it is true that mutations observed in the first codon position are more significant than mutations observed in the third codon position, the use of DNA data is less arbitrary and subjective than the choice of which morphological features to include in a set of characters in a study.

On the other hand, there are problems with biological sequence comparisons too. To compare two sequences, most algorithms require that they first be aligned. The very act of aligning sequences requires ignoring information contained in the sequences. The alignment of a sequence with two adjacent identical characters, e.g. the character "a", to a second sequence with one of those positions deleted is arbitrary: either of the two characters could map to the undeleted character in the second sequence. Compare this to a third sequence where the adjacent characters "a" and "c" are mapped to the second sequence; the choice is quite clear. But in both cases the same choice, the deletion of the second position, will be made by an arbitrary alignment. Information is lost. There are now standard alignment methods applied to sequences (See Kruskal and Sankoff 1983). Good alignment methods can reduce errors, but no alignment method can eliminate them.

Another difference between morphological and sequential data is that morphological features are fixed in place by evolution. Many morphological features are functional. But observed mutations in sequences are not necessarily functional. Not all differences in biological sequences result in a discernible difference in the function of an organism. Mutation is a process of making mistakes. Evolution is a process of figuring out which mistakes are not the wrong ones. Sequential data sets, especially those which represent biological sequences that may be noncoding or may have synonymous mutations, accumulate differences more easily than data that require the fixation of a trait to define an OTU.

In 1969 Jukes and Cantor improved upon the Hamming distance. Realizing that more than one substitution could take place at the same site, Jukes and Cantor devised a corrected distance to estimate the expected number of actual substitutions, based on the number of observed substitutions and sequence length. The method still uses a parsimonious alignment as input, however, and suffers from the failings of parsimony, as will be described in chapter two. The original "one parameter" Jukes-Cantor method has led to many refinements such as the two-parameter method based on transitions (changes from one purine to the other or one pyrimidine to the other) and transversion (changes from a pyrimidine to a purine or vice-versa) probabilities (Kimura, 1980), the three-parameter method based on the transition and two transversion probabilities (Kimura 1981), a four-parameter model that takes ratio of AT to GC content into account (Takahata and Kimura 1981) and a sixparameter method which takes the content of each nucleic acid base into account (Gojobori et al 1982).

Although they have not received as much attention as the Jukes-Cantor type distances, models based on the variability of codon sites are useful when a known reading frame exists for a set of DNA sequences. Under the neutral mutation hypothesis (Kimura 1983), that most mutations are neutral with respect to biological function, synonymous mutations (those that do not change the amino acid coded for by a codon) do not suffer the constraints of selection. They occur more often than nonsynonymous mutations. Perler et al. (1980) categorized each possible visible and silent mutation by the number of synonymous and nonsynonymous event possibilities available at a given site for the apparent codon. The order of events when two or three mutations appear within one codon is unknown so a fractional value could be defined and distributed among more than one of these categories, e.g. the codon change for CUC to UUA could involve two synonymous events, CUC (leucine) to UUC (phenylalanine) to UUA (leucine), or three synonymous events, CUC (leucine) to CUA (leucine) to UUA(leucine). A Jukes-Cantor like correction is then made for each category of substitution. Miyata and Yasanuga (1980) expanded the definition of synonymous by incorporating the degree of acceptance for a visible mutation, e.g. a change from valine to leucine is more "synonymous" than a change from valine to aspartic acid. Rather than split the contribution of a multiple event codon evenly among categories, a weighting scheme is implemented based on the degree of synonymity. Nonsynonymous changes were not categorized. Nei and Gojobori (1986) found that Miyata and Yasanuga's method gives essentially the same estimates of distance when the weighting scheme is not used.

Building Phylogenetic Trees

One of the primary uses of distance measures, molecular or morphological, is for building phylogenic trees (phylogenetic when using DNA). In 1967 Fitch and Margoliash created phylogenies based on amino acid sequence differences. A nearly identical method was introduced independently by Cavilli-Sforza and Edwards (1967). Under the Fitch-Margoliash method, the number of DNA mutations required to obtain the observed amino acid mutations is used to create a matrix of sequence distances. More mutations corresponds to greater distance. The method is used to create trees. Parsimony is used here in a different form: it is assumed that the correct topology of the phylogenetic trees is that which minimizes the total distance between all OTUs in the phylogeny. This process maximizes overlap of pathways along branches leading to multiple OTUs. It is simple enough to join three OTUs so long as they do not violate a triangle inequality. But when more than three OTUs exist it may not be possible to find a position for an additional OTU which arithmetically agrees with all the matrix distances between that OTU and the joined OTUs. In such cases the closest OTU pair is joined and then treated as a single OTU, averaging the distances between the clustered OTU and the remaining OTUs in the new distance matrix. The process is repeated, lowering the number of OTUs with each iteration until all are joined.

Other methods exist for converting sequence data to phylogenies using distance matrices. Described below are some of the more popular: the neighbor joining method (Saitou and Nei 1987), the unweighted pair group method (see Sneath and Sokal 1973) and the modified (Tateno et al 1982) Farris method (Farris 1972). Methods of this latter type do not build phylogenies by clustering OTUs together, treating them as new OTUs, and condensing the phylogeny to one large cluster. Instead they use a nucleation algorithm. Starting with a core of the two closest OTUs, successive OTUs are added until the total phylogeny is built. The choice of the next OTU at each step is based on the potential branch length between each remaining OTU and the "already joined" OTU nearest each potential branch point of the tree. Potential branch lengths for all unjoined OTUs and all potential branch points are calculated before adding the chosen OTU to the tree.

The Unweighted Pair Group Method - Arithmetic Average (sometimes called UPMGA) system is similar to the method of Fitch and Margoliash except that once a cluster has been redefined as an OTU the distance between a composite OTU and another OTU is based on the unweighted average distance between all members in the composite OTU. For example, if ***a***, ***b***, ***c*** and ***d*** represent OTUs and d_{xy} is the distance between OTUs, then the distance between the composite OTU ((***a***, ***b***), ***c***) and the OTU ***d*** is computed as $d_{ad}/3 + d_{bd}/3$

+ $d_{cd}/3$ rather than $d_{ad}/4 + d_{bd}/4 + d_{cd}/2$. This is simpler than the neighbor joining method, which uses all the sequence information. The neighbor joining method treats the set of OTUs, at the outset, as a star topology with all OTUs around one center node *X*. A cluster of the two closest OTUs is separated by moving all the remaining OTUs to a new node *Y*. The distances between all pairs of OTUs are used to calculate the distance between the two removed clusters, the node *Y*, and the node *X*. *X* and *Y* are dynamic. That is, once the two separated OTUs are joined to form a new composite OTU, *Y* becomes the center node *X* and the process is repeated. In both the neighbor joining method and UPGMA we are left with one less OTU each time we cycle through the process; eventually all OTUs are joined to a single tree (or equivalently all nodes are reduce to three branches).

Because a loss of information accompanies distance measures, construction of phylogenies directly from data is preferred. Tateno (1990) observed that "In the construction of molecular trees, the sequence data are more informative than the distance matrix". The point is that when a phylogeny is built directly from the data, rather than through an intermediate distance, there is less loss of potentially useful information. The two most widely used methods of building phylogenies directly from sequence data have created some controversy. The method which has been in use for the longest time, Maximum Parsimony (MP), is easy to calculate. Maximum Likelihood (ML) methods are computationally more expensive but reflect knowledge of biochemistry through model based assumptions about sequence mutation.

When applied to phylogeny, the parsimony principle assumes that the topology which minimizes the number of steps (mutations) connecting the observed sequences will produce the best phylogenetic tree. Wagner (1961) invoked parsimony to analyze phylogeny using nonsequential morphological data; that method was modified by Fitch (1971) for use with biological sequences. Today the best results are found (Huelsenbeck and Hillis 1993, Hillis et al 1994) using Weighted Parsimony (Sankoff 1975, Swofford and Olsen 1990), a modification of MP where the identity of mismatched characters is used to calculate branch length rather than just a count of mismatched characters.

Parsimony as applied to molecular phylogeny has been described as "questionable at best" (Felsenstein 1988), not truly having a stochastic model that consistently obtains the correct tree (DeBry 1992, Sidow 1994) and converges on the wrong tree when given enough data from systems where branches have disparate rates of mutation per unit time (Cavendar 1978, Felsenstein 1978). But no method can claim to always converge on the correct tree and parsimony-based phylogenetic packages (Maddison and Maddison 1992, Swofford 1992) enjoy continued use (Stewart 1993). Parsimony-based phylogeny methods have been tested against theory (Goldman 1990), biological data (Atchley and Fitch 1991, Hillis and Bull 1991, Hillis et al. 1992, Bull et al. 1993), and computer simulation (Hendy and Penny 1989, DeBry 1992, Hasegawa and Fujiwara 1993, Hillis and Swofford 1994). Debate continues over the accuracy of biological data (Sober 1993, Hillis et al 1993) and the choice of model for computer simulation testing methods (Hillis et al 1994). Parsimony-based methods have been favorably compared to more computationally intensive methods (Huelsenbeck and Hillis 1993, Hillis et al 1994).

Maximum Likelihood (Felsenstein, 1981) systematically determines the relative probability of obtaining the events necessary to produce each step in a given topology of OTUs. There are two parts to ML: propagation of probability for an event to affected OTUs, based on a tree topology, and probability of observing the events necessary to create that tree topology. The first part is an organization of summations and "prior state probabilities" so that the equation for calculating the likelihood of a tree accurately reflects the topology of the tree. The second part is the calculation of the prior state probabilities for the equation, a Markovian analysis of the probability of observing the known characters of the OTUs assuming all possible common ancestral characters. A “likelihood” of realizing a given tree is obtained. This provides a measure for comparison between possible phylogenies. There is a limitation: not all possible tree topologies can be calculated in a reasonable time for higher numbers of OTUs. While alignment is necessary for the original ML method, an ML based method which allows insertions and deletions has been presented (Thorne et al 1992).

Chapter 2

An Exact Simulation of Sequence Mutation to Study Parsimony and Distance Measurement

Abstract

We evaluate the parsimony principle as it pertains to measures of "similarity" or "distance" between sequences of DNA or protein molecules. The parsimony concept and its corresponding distance measures (e.g. Hamming or Levenshtein distances, see Kruskal 1983) assume that the relatedness of two sequences can be determined by the *shortest path*, the minimum number of modifications needed to mutate one sequence into the other. But true sequence evolution, like processes of chemical diffusion, need not follow the shortest path. Our computer model begins with a given parent sequence of symbols and, through complete enumeration of successive generations of every possible substitution, insertion, and deletion follows every possible "pathway" from parent sequences to daughters. We find situations where parsimony gives incorrect rank orderings of the closeness of one parent/daughter pair relative to another such pair. In particular, sequences that are *homogeneous* (predominantly composed of a single character) can evolve through a larger number of pathways than sequences that are heterogeneous (having a broader composition of characters), implying that monomer placements in the sequence are not independent of their neighbors with respect to evolutionary relatedness.

Introduction

We develop a simple computational model of the evolution of sequences of symbols, such as DNA or proteins. Our aim is to study the principle of parsimony, the idea that "distances" or sequence dissimilarities are given by the shortest mutational paths, i.e., the smallest number of mutations that change one sequence into the other. By using exact computer enumeration, we study all the possible mutational paths, not just the shortest paths. In this way, we test the parsimony assumption as it applies to molecular distance.

While parsimony underlies some distance measures, its most common usage is in the creation of phylogenetic trees. Parsimony was first used with biological sequences for purposes of phylogeny by Eck and Dayhoff (1966). The justification for parsimony is based more on convenience than on principle: determining a shortest mutational pathway is simple and unambiguous. But Eck and Dayhoff recognized that the evolutionary road may well have involved superimposed substitutions or such superfluous pairs of events such as an insertion later undone by a deletion at a particular site on a DNA or protein sequence. Such complementary pairs of mutations are neglected by the parsimony assumption, which treats these instances as if they involved zero mutations. In this way parsimony neglects the many possible real paths evolution could have taken from a parent to a daughter sequence.

The parsimony principle is applied to phylogeny through the assumption that the best phylogenetic tree has branches defined by minimal numbers of mutational steps between sequences. Wagner (1961) invoked parsimony to analyze phylogeny using morphological data. Fitch (1971) modified that method for use with biological sequences. Today the best results are found (Huelsenbeck and Hillis 1993, Hillis et al 1994) using Weighted Parsimony (Sankoff 1975, Swofford and Olsen 1990), a modification in which branch lengths depend not only on the number of mutations but also on the specific characters involved. Parsimony as applied to phylogeny has been described as "questionable at best" (Felsenstein 1988): it is not based on an underlying stochastic model that consistently gives the correct tree (DeBry 1992, Sidow 1994), and it converges on the wrong tree when given enough data from systems where branches have different clock rates (Cavendar 1978, Felsenstein 1978).

On the other hand, no method currently provably converges on the correct tree and parsimony has the advantage of ease of calculation leading to the continued use (Stewart 1993) of phylogenetic computer algorithms (Maddison and Maddison 1992, Swofford 1992). Parsimony-based phylogeny methods have been tested against theory (Goldman 1990), biological data (Atchley and Fitch 1991, Hillis and Bull 1991, Hillis et al. 1992, Bull et al. 1993), and computer simulation (Hendy and Penny 1989, DeBry 1992, Hasegawa and Fujiwara 1993, Hillis and Swofford 1994). Debate continues over the accuracy of biological data (Sober 1993, Hillis et al 1993) and the choice of model for computer simulation testing methods (Hillis et al 1994). Parsimony-based methods have been compared to more computationally intensive methods (Huelsenbeck and Hillis 1993, Hillis et al 1994). These efforts are important for leading to better understanding of evolutionary relationships.

Although the "four taxon" problem (figure II-1) is of more direct importance to phylogeny, our present work is restricted by computer limitations to the modest goal of studying parsimony as applied to simple parent/daughter distances. As noted by Yang (1996), it is helpful to identify the underlying factors that account for a method's successes or failures.

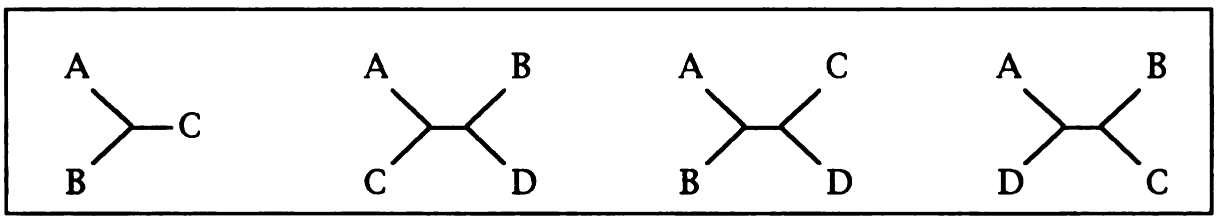


Figure II-1 The 4 taxon problem. With three OTUs, only one branch point is necessary. With four OTUs, the placement of branchpoints must be solved. Three possible configurations for the four OTUs A, B, C and D are presented in the latter three diagrams.

The Model

Our aim here is to explore a simple computer model of sequence evolution. We consider an alphabet of k characters (a, b, c, ...). For example k = 4 nucleotides represents DNA molecules or k = 20 amino acids represents proteins. We begin at time $t = 0$ with a parent sequence, labeled P_0. At each tick of an event clock, our computer algorithm inserts, deletes, and substitutes every possible character at every possible site to create an ensemble of daughter sequences. We call this ensemble the "daughter space". The first-generation daughter sequences are then used as parent sequences to produce the next generation, t = 2, of daughters, which then serve as the parent sequences for the next generation, t = 3, and so on. There can be multiple "pathways"; alternative series of mutational events that lead to a given daughter sequence. The extent to which a daughter sequence appears in daughter space is determined by two factors: (i) the mutational frequencies of the individual characters (for example, if insertions are more frequent than deletions, then longer daughter sequences will be more likely than shorter daughters), and (ii) the number of mutational paths from parent to daughter. We define a daughter sequence's *statistical weight* as the relative portion of sequence space taken by that sequence.

A principal tenet of the present work is that the evolutionary distance between two sequences is not the length of the shortest path; it is an average over all the possible paths. Thus evolutionary distance is not just a matter of the length of the minimal path, the basis for parsimony-based distance measurement. Rather, evolutionary distance also depends on the numbers of paths. Two sequences A and B may have a "closer" relationship than sequences C and D by virtue of a larger number of paths from A to B for the same given number of mutational events. In this sense we believe evolution mimics diffusion, where the time required for particles to reach a particular point is better described by ensemble averages than by shortest paths.

Our model is not intended as an accurate model of biological evolution, with all the complexities of biological machinery that can bias how mutations are made and kept. Ours is simply a model of principle. Recognizing that real evolution follows paths that need not be minimal, we simply use this idealized model to ask: what are the errors incurred in neglecting the multi-pathway nature of evolutionary processes and approximating them as shortest paths?

Sequence generation

Here we describe the generation method and define some terms. The terms "parent" and "daughter" define the arrow of time in the model for a relationship across any number of generations. "Immediate daughter" and "immediate parent" specify a relationship across one generation. A sequence is denoted by a string of characters in braces, e.g. {a b c}, where a, b and c represent characters from the given alphabet. A relationship between daughter sequence D and parent sequence P is written D|P, e.g. "{a b a} as a daughter of {a a a}" is written as {a b a}|{a a a}. Only one event can occur at each tick of the clock: a substitution, deletion, or insertion. The relative frequencies are given by f_s, f_d, and f_i, respectively. The weighted daughter space, $D_w(P_0,t)$, is the set of all daughters obtained after t mutations to the given parent sequence, P_0. Therefore $D_w(P_0, 0)$ is the space containing the single parent sequence at time $t = 0$. The statistical weight for P_0 is 1.

To create the jth generation, $D_w(P_0, j)$ from generation j-1: i) all possible substitutions, deletions and insertions are made on every sequence of $D_w(P_0, j\text{-}1)$ and ii) any sequence that appears more than once is pooled and its statistical weights summed. The statistical weight for a daughter sequence therefore arises from two factors: the intrinsic weight due to the event frequencies and the number of paths from parent to daughter. Multiple pathways to a given daughter are called "degenerate" pathways, and daughter sequences that appear many times are called degenerate sequences.

The model has three principal assumptions:

1) Equiprobability of characters. Every character in the alphabet is substituted, deleted, or inserted with the same probability as every other. A substitution event may involve replacement of a character x by an identical character x. Character probabilities are stationary and symmetric (see Lockhart et al 1994) over all generations from a parent sequence to a daughter sequence.

2) No sequence site is special. The mutation frequencies are independent of the mutation site in the parent sequence. This is an idealized model that neglects codon placement or conservation due to evolutionary pressure (see Palumbi 1989; also Shoemaker and Fitch

1989). A sequence site that is modified by a substitution or created by an insertion event is fully susceptible to a subsequent event.

3) Mutation sites are independent. The probability or type of mutation is not affected by the characters in the adjacent sites. For example, a thymine next to a thymine is creates a spot that is prone to mutation due to thymine dimerization (see Ayala and Kiger 1980). The rate and type of mutation is not affected by sequence composition.

The Original Parent Sequence is a Subsequence of a Longer String

To establish how to treat certain end effects, we assume that parent sequence P_0 is contained within a larger genome. That is, the parent is bounded by a symbol *x* on the left and a symbol *y* on the right; these symbols are immutable during the computer evolution. The original parent sequence is bound by positions to the left and right which are assumed to be conserved and immutable. The need for such a boundary condition arises because we need to define what constitutes a match between a parent and daughter. We assume the daughter too is bounded by *x* and *y*. Therefore if we are interested in some property of a daughter sequence {*x* a a a *y*} then a supersequence such as {*x* b a a a b *y*} is not counted as contributing to the statistical weight.

If the number of generations is larger than the number of positions in the sequence then it is possible for a sequence to incur enough deletions to reduce the sequence to zero length. Zero length sequences are allowed to have insertions (substitutions or deletions are not possible) in the following generation because positions *x* and *y* have not disappeared and define a point of insertion. Unlike Bishop and Thompson (1986) or Thorne et al. (1992), we do not assume insertion probabilities at the ends of a daughter sequence to be one half or take place at only one end of the sequence due to the presence of the supersequence.

Example

Table II-1 shows an example computer simulation of sequence mutation: $D_w(\{a\ \ a\ \ c\},1)$ is the first generation obtained from the parent {a a c}. Weights are calculated on a per position basis (as discussed later) based on the event type:.

1) Each substitution creates a new daughter with a statistical weight $w_d = w_p f_s/k$ where w_p is the weight of the immediate parent.

2) Each deletion creates a new daughter with a statistical weight $w_d = w_p f_d$.

3) Each insertion creates a new daughter with a statistical weight $w_d = w_p f_i/k$.

Substitution	w_d	Deletion	w_d	Insertion	w_d
{**a** a c}*	1/12	{a c}*	1/3	{**a** a a c}*	1/12
{**b** a c}	1/12	{a c}*	1/3	{**b** a a c}	1/12
{**c** a c}	1/12	{a a}	1/3	{**c** a a c}	1/12
{**d** a c}	1/12			{**d** a a c}	1/12
{a **a** c}*	1/12			{a **a** a c}*	1/12
{a **b** c}	1/12			{a **b** a c}	1/12
{a **c** c}	1/12			{a **c** a c}	1/12
{a **d** c}	1/12			{a **d** a c}	1/12
{a a **a**}	1/12			{a a **a** c}*	1/12
{a a **b**}	1/12			{a a **b** c}	1/12
{a a **c**}*	1/12			{a a **c** c}*	1/12
{a a **d**}	1/12			{a a **d** c}	1/12
				{a a c **a**}	1/12
				{a a c **b**}	1/12
				{a a c **c**}*	1/12
				{a a c **d**}	1/12

Table II-1 The immediate daughter sequences of P_0 = {a a c} with $f_s = f_d = f_i = 0.333$ and the character set (a, b, c, d): $k = 4$. * denotes a degenerate sequence: it appears more than once in the table but once with a total weight in the weighted daughter space. Bold indicates the changed character for insertions and deletions. Below is the daughter space D_w({a b c}, 1) with weights.

{a a c}*	1/4,	{b a c}	1/12,	{c a c}	1/12,
{d a c}	1/12,	{a b c}	1/12,	{a c c}	1/12,
{a d c}	1/12,	{a a a}	1/12,	{a a b}	1/12,

{a a d}	1/12,	{a c}*	2/3,	{a a}	1/12,
{a a a c}*	1/4,	{b a a c}	1/12,	{c a a c}	1/12,
{d a a c}	1/12,	{a b a c}	1/12,	{a c a c}	1/12,
{a d a c}	1/12,	{a a b c}	1/12,	{a a c c}	1/6,
{a a d c}	1/12,	{a a c a}	1/12,	{a a c b}	1/12,
{a a c d}	1/12.				

Results

Kinetic Accessibility: A Consequence of Multiple Paths

A particular daughter sequence D_X arises from a given parent sequence P_0 according to a time distribution function. We define $w(D_X, D_w(P_0, t))$ to be the weight of the probe sequence D_X in the daughter space $D_w(P_0, t)$). Similarly, $p(D_X, D_w(P_0, t))$ is the fraction of all possible daughter sequence comparisons that successfully identify D_X:

$$p(D_X, D_w(P_0, t)) = \frac{w(D_X, D_w(P_0, t))}{\sum_{D_n \in D_w(P_0, t)} w(D_n, D_w(P_0, t))} \tag{1.1}$$

where D_n represents any daughter sequence n in the daughter space.

One of the main points of this paper is to introduce the kinetic accessibility of a daughter sequence from its parent. High accessibility means that a daughter sequence can arise from the parent through either a greater number of different mutational pathways and/or the use of higher probability events. Low accessibility means there are fewer and/or less preferred pathways. While the probability defined by eq. 1.1 is normalized over all daughter sequences that appear in event cycle t, our interest in the distance between parent and daughter sequences requires an additional normalization. We seek the expected time $\langle t \rangle$ at which the daughter sequence D_X is observed in the daughter space of P_0 (eq.

1.2). The kinetic accessibility describes *when* the sequence is observed, on average. Therefore we now normalize over generations to find the expected event cycle t in which the sequence will appear:

$$\langle t(D_X|P_0)\rangle = \frac{\int_{t=0}^{\infty} t \bullet p(D_X, D_w(P_0,t))dt}{\int_{t=0}^{\infty} p(D_X, D_w(P_0,t))dt} \tag{1.2}$$

We take this quantity <t> as a definition of the evolutionary distance between a parent and daughter sequence. Because it is not computationally practical to cover an infinite number of event cycles and our time steps are discrete, we approximate eq. 1.2 with:

$$\langle t(D_X|P_0)\rangle = \frac{\sum_t t \bullet p(D_X, D_w(P_0,t))}{\sum_t p(D_X, D_w(P_0,t))} = \sum_t t \bullet A(D_X, D_w(P_0,t)) \tag{1.3}$$

where $A(D_X,D_w(P_0,t))$ is the kinetic accessibility and the time t is summed over all generations of daughter space available:

$$A(D_X, D_w(P_0,t)) = \frac{p(D_X, D_w(P_0,t))}{\sum_t p(D_X, D_w(P_0,t))} \tag{1.4}$$

In the section below, we compare such distances from parent sequences to daughter sequences. On the one hand, we consider parsimony-based shortest-path distances which are direct counts of the minimum number of mutations required to convert one to the other. We compare that measure of dissimilarities to the kinetic accessibilities and <t>, the *average* number of mutations used to convert the parent to the daughter. We find cases in which parsimony-based measures would find that two parent/daughter relationships $P_1|D_1$ and $P_2|D_2$ have the same shortest pathlength while the accessibility measure shows their relationships to be quite different. The figures below show accessibilities vs. generation number *t*. The shape of the accessibility profile is a determinant of the closeness of a parent to a daughter. To minimize arbitrariness, the profiles below are computed using equal rates of substitution, insertion, and deletion: $f_s = f_d = f_i$.

Accessibilities

Figure II-2B shows one of the main results of this chapter, an example where parsimony-based distance can err in rank ordering the relative closeness of two parent/daughter relationships. We take the simplest case of two relationships, each daughter being identical to its own parent. D_1 is {a a a a a a}, a daughter of parent P_1: {a a a a a a}, and D_2 is {a b c d a b}, a daughter of parent P_2: {a b c d a b}. The parsimony-based Hamming distance between two aligned sequences is the count of the number of differences, position-by-position. According to parsimony, both the parent/daughter relationships above are equally close because both have Hamming distances equal to zero. But figure II-2B shows that the accessibilities are significantly different for these two parent/daughter pairs. In all generations beyond the first the sequence {a a a a a a} is more accessible from its parent than {a b c d a b} is from its parent.

The difference is due to a property of the composition of characters in the sequence that we call *homogeneity*, which is not taken into account by parsimony-based methods. Sequence {a a a a a a} defines perfect homogeneity: all the characters are identical. In contrast sequence {a b c d a b} is heterogeneous: the composition of symbols is more diverse. More pathways connect pairs of homogeneous sequences than pairs of heterogeneous ones. This is because a deletion of the character "a" in the sequence {a a a a a a} at position 1 can be followed in a subsequent generation by insertion of an "a" at *any* position to produce the given daughter. But in the heterogeneous sequence a deletion at position 1 can only be followed by an insertion at position 1 to produce the given daughter. The same principle holds true for an insertion of the character "a" followed by a deletion in a subsequent generation.

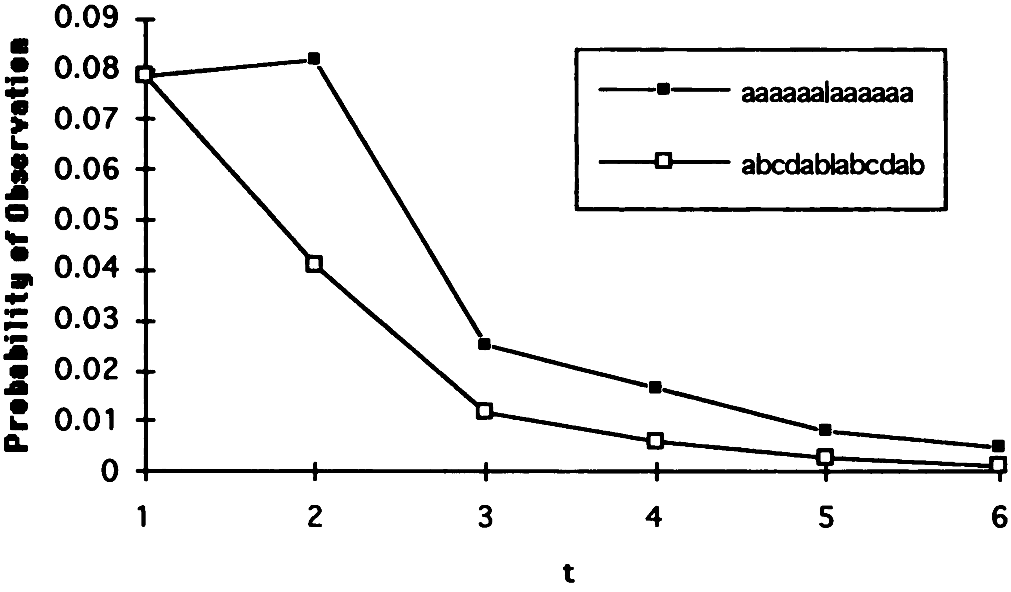
Hence there are more paths connecting homogeneous parent/daughter pairs than heterogeneous pairs. This is an example of how kinetic accessibility differs from parsimony. By definition, parsimony takes no account of any sequence composition effects. Using a thermodynamic analogy, parsimony resembles energy, and neglects entropy, the number of different ways of achieving a given energy. The real driving forces in nature are described by the free energy, a combination of energy and entropy. In this sense, kinetic accessibility is more akin to a true free energy, and parsimony-based measures are akin to using energies to approximate free energies.

Kinetic accessibility introduces the analog of entropy into sequence comparisons.

One consequence of following all paths, rather than just the minimal path, is that the expected distance between a parent sequence and its identical daughter sequence will not be zero as the mutational clock continues to tick. For {a a a a a a}||{a a a a a a}, the expected distance is 2.1 t while for {a b c d a b}||{a b c d a b} the expected distance is 1.7 t.

Because this result is counterintuitive, given that homogeneity leads to more paths to a daughter sequence, it calls for explanation. This result is a trivial consequence of normalization. Both pairings have the same probability in the first generation because only the substitution of a character by itself will produce the given daughter of either parent. But because homogeneous parent/daughter pairs are connected by more pathways at all later generations (fig. II-2A), the later probabilities are higher, but when normalized to 1 over all generations, the homogeneous pairing distribution will be uniformly reduced (fig. II-2B), hence the result above.

A



B

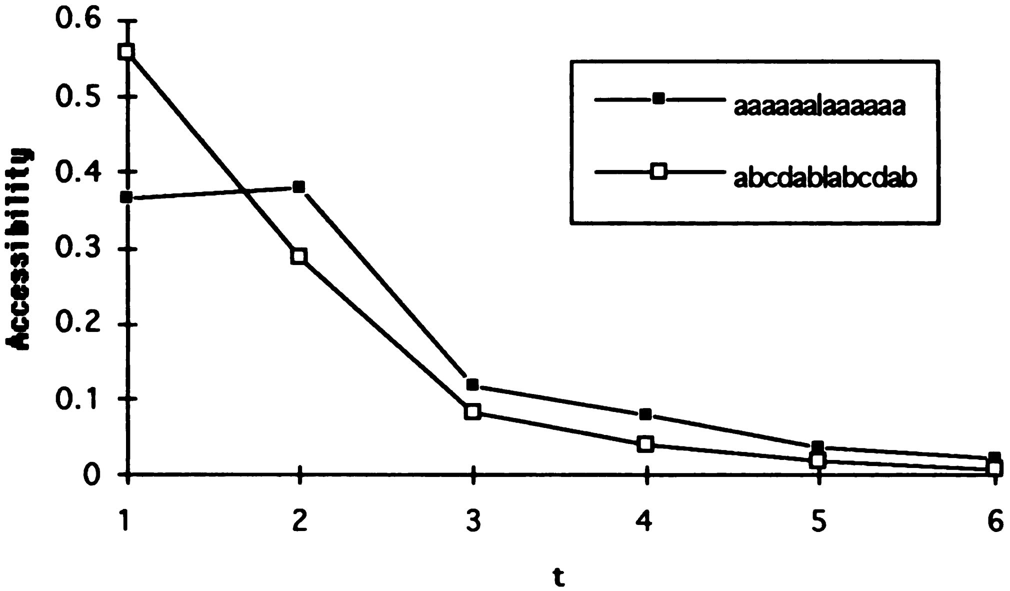


Figure II-2 A The probability of observing of {a a a a a a}|{a a a a a a} as a function of time and the probability of observing of {a b c d a b}|{a b c d a b) as a function of time. **B** The accessibility of {a a a a a a}|{a a a a a a} as a function of time and the accessibility of {a b c d a b}|{a b c d a b) as a function of time. k = 4.

Figure II-3 shows another example of an error in approximating an expected distance by a shortest path. This example is more general in that it includes insertions and deletions. When insertions and deletions are involved, parsimony requires a generalization of the Hamming measure called the Levenshtein distance. The Levenshtein distance treats inserted or deleted positions as mismatches. Figure II-3 shows parent/daughter pair {a a a}||{a a a a a a}, which has a Levenshtein distance of 3 and {b a a a a c}||{a a a a a a}, which has a Levenshtein distance of 2. Therefore, according to parsimony, the latter parent/daughter relationship is closer. However, the average calculated distance for the first pair is 3.82 t and for the second is 3.97 t. The latter daughter emerges from its parent later than the former daughter does from its parent. The shapes of the accessibility profiles demonstrate the preference for the former relationship: the former pair is more accessible at generation 3 then less accessible at later times.

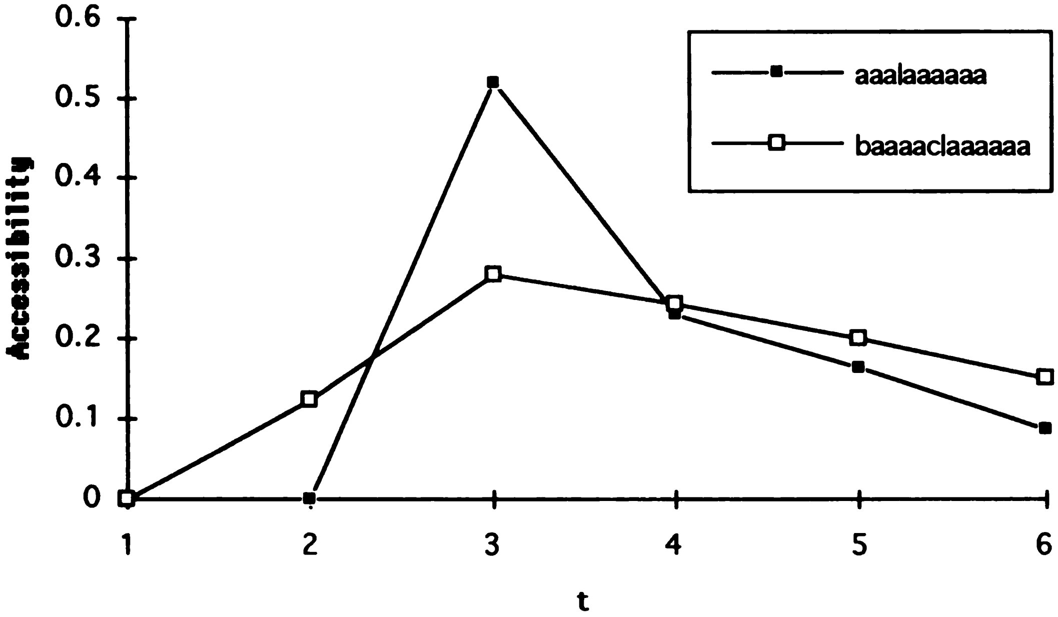


Figure II-3 The accessibilities of {a a a}|{a a a a a a) and {b a a a a c}|{a a a a a a) as functions of time. **k** = 4.

Figure II-4 shows a different test of parsimony. Using all sequences of length 6, we have collected all the parent/daughter pairs having a Hamming distance exactly equal to one (i.e. differing by one mutation). Figure II-4 is a histogram of the accessibility-based expected distances among this set of parent/daughter pairs. If parsimony were exactly correct, figure II-4 should have a single peak. Instead, the breadth of this distribution indicates evolutionary variations missed by the Hamming measure.

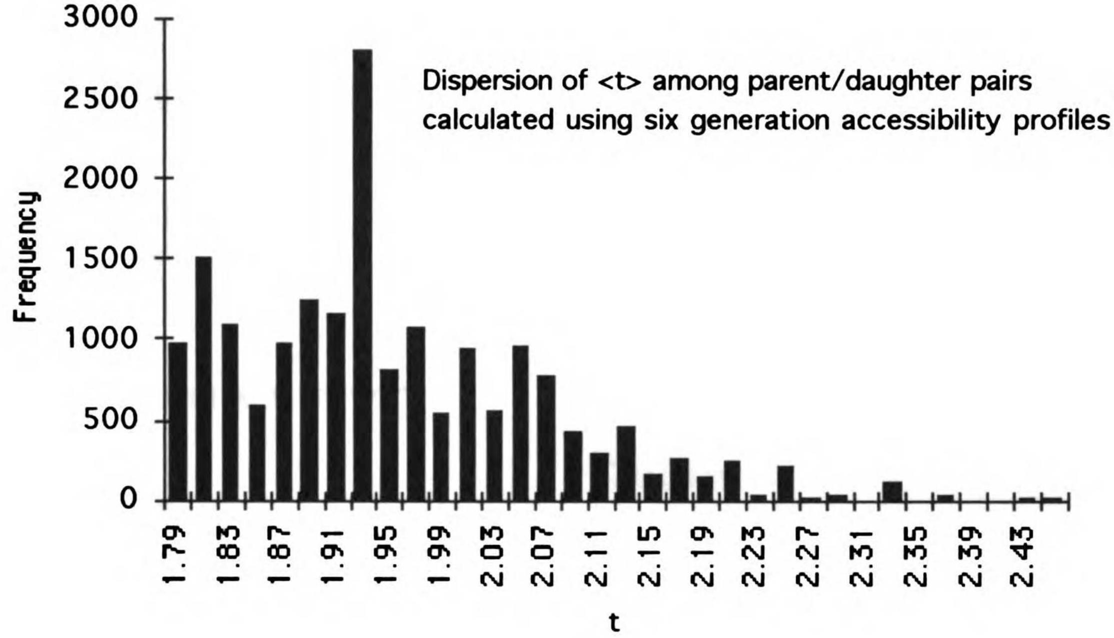


Figure II-4 The distribution of expected distance for all possible parent/daughter relationships having a Hamming distance of one using sequences of length 6. k = 4.

Convolution: Another measure of relatedness

The expected value, <t>, is only one measure of a time distribution function. Other measures could also be chosen to define the relatedness between two sequences. Now rather than use the mean value $<t>$ to determine the relative closeness of two relationships, we ask: Which of two parent/daughter relationships is more likely to be separated by fewer event cycles? This is a measure of closeness that uses all the information in the distribution functions between two parent/daughter relationships, rather than using only the single quantity <t>.

$R(A < B)$, the precedence of A over B, is defined as the probability that parent/daughter relationship A is closer than parent/daughter relationship B:

$$R(A<B)=\sum_{i=2}^{t_{\max}}\left(A(D_b,D_w(P_b,i))\sum_{j=1}^{i-1}A(D_a,D_w(P_a,j))\right) \tag{2.4}$$

or for the reverse situation,

$$R(B<A)=\sum_{i=2}^{t_{\max}}\left(A(D_a,D_w(P_a,i))\sum_{j=1}^{i-1}A(D_b,D_w(P_b,j))\right) \tag{2.5}$$

We neglect cases in which both relationships have identical distances ("dead heats") therefore $R(A < B) + R(B < A)$ may sum to less than one.

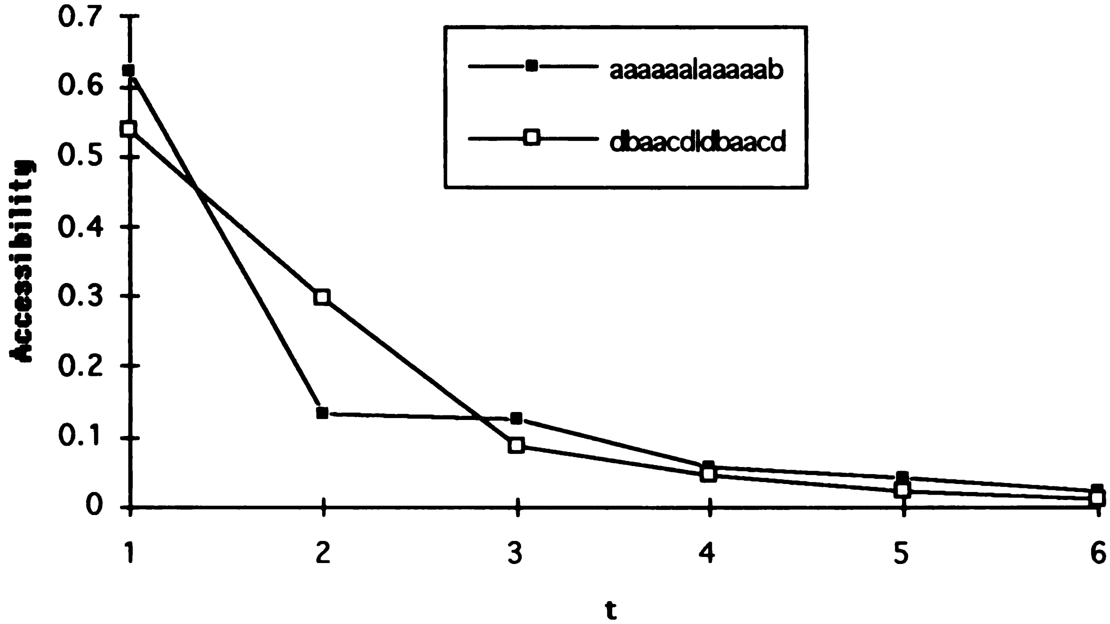


Figure II-5 The accessibility of {a a a a a a}|{a a a a a} as a function of time and the accessibility of {d b a a c d}|{d b a a c d} as a function of time. k = 4.

Relation\\t	1	2	3	4	5	6
A	0.62046	0.13415	0.12526	0.05741	0.03971	0.02297
B	0.53865	0.29756	0.08651	0.04541	0.02060	0.01123

Table II-2 The accessibility of A, {a a a a a a}|{a a a a a}, and the accessibility of B, {d b a a c d}|{d b a a c d}, for generations 1 to 6. k = 4.

Figure II-5 records an instance in which both <t> and the Levenshtein distance show that relationship A, {a a a a a a}|{a a a a a}, is more distant than relationship B, {d b a a c d}|{d b a a c d}. But the precedence quantity R disagrees with both of them. The expected distance for A is 1.83 t and for B is 1.745 t, but R(A < B) = 0.320 while R(B < A) = 0.291

(see table II-2). The implication is that time distributions can be complex. Not only do shortest path approximations sometimes err, but also other various simple measures are not always unambiguously correlated with each other.

Conclusions

We have presented a simple computer algorithm that generates all possible mutational pathways from a parent sequence of symbols to all daughter sequences. Because this study involves exact computer enumeration, it is not a practical algorithm for sequence comparisons and is limited to short chain lengths. This is simply intended as a model study of how evolutionary processes can follow different mutational pathways and how the numbers of mutational routes can contribute to the relatedness of two different sequences.

We find that distance measures based on parsimony — shortest mutational paths — can err in predicting the greater similarity between two parent/daughter relationships, even following strict assumptions (constant molecular clock, equiprobable characters). The main implication is that the relatedness between two sequences is not just a matter of character-by-character counts of mismatches, as if the probability of obtaining a sequence was independent of the total sequence. Rather, relatedness also depends on the overall composition of characters — the relative populations of each symbol type, and also whether a symbol in the string is identical to the symbol of a position nearby. This result also bears on methods that extrapolate distance measures from parsimonious measures (e.g. Jukes and Cantor 1969, Kimura 1980).

Nevertheless, two aspects of our model are reflected by parsimony as applied to distance measurement. First, the shortest path may often be a reasonable approximation to the accessibility time distribution functions which invariably converge to zero as the number of generations increases. That is, kinetic accessibility is highest in the early generations, even within a model which has no generational bias. Second, kinetic accessibility requires a normalization over all generations to separate the probability of observing a sequence from the probability that an observation is made at time t. Parsimony, in its simplicity, involves an implicit normalization over one data point.

Chapter 3

How Mutation Evolves Order and Disorder in Biomolecular Systems

Abstract

We consider the process by which a parent sequence evolves to an ensemble of daughter sequences through an exhaustive set of every possible substitution, deletion, and insertion. "Daughter space" can be thought of as a multidimensional grid, with each node representing a daughter sequence, and each connecting link representing a mutation. Each node carries a "degeneracy", representing the frequency of appearance of that daughter. We find that daughter space looks much like cities on a roadmap. Just as there are large hub cities interconnected with small towns, daughter space has sequences of high degeneracies and low degeneracies. The evolutionary process is like the flow of traffic through this grid. While greater similarity of a daughter sequence to its parent sequence results in higher degeneracy, another contributor to degeneracy is sequence "homogeneity". A perfectly homogeneous sequence is one having all identical characters (e.g. {a a a ...}); a heterogeneous sequence has all different characters (e.g. {a b c ...}). The degree of homogeneity provides an "order parameter", which we use to follow the kinetics of evolutionary change. Such ordering and disordering processes imply an arrow to evolutionary time.

Introduction

Our aim here is to explore a model for the mutational processes by which a parent sequence mutates to its daughter sequences. The set of all sequences with a given character set can be viewed as a multidimensional lattice. The sequences constitute the nodes of this lattice. The connections between the nodes represent mutational events: substitutions, deletions and insertions. The possible evolutionary routes interconnecting any two sequences can be determined by tracing the possible paths through intervening sequences on this multidimensional grid. Starting from a given parent sequence P_0, which is one point on this lattice, our computer algorithm systematically finds all daughter sequences that are one step away, then two steps away, etc.

We define two properties: the "degeneracy" of a daughter from a given parent, and the "homogeneity" or "heterogeneity" of a sequence. A degenerate sequence is a daughter sequence that can be obtained using multiple mutational paths from the parent sequence. The number of evolutionary paths from the parent defines the level of degeneracy of the daughter. Homogeneity and heterogeneity define a property of the composition of a given sequence. If a sequence is a string of identical symbols, {b b b b}, it is perfectly homogeneous; if the characters are all different, {a b c d}, it is heterogeneous. This measure of composition provides an "order parameter" - homogeneous sequences are highly ordered and heterogeneous sequences are disordered. We explore the time evolution of ordering and disordering of sequences in our model.

In this chapter, we ask two questions. First, how "clustered" are the nodes in this daughter space? Are there "favored" daughter sequences? Parsimony-based distance measures assume that all first-generation daughters are equally populated. We find here that they are not. Second, what is the time evolution of order and disorder as a parent sequence mutates to its ensemble of daughter sequences? Homogeneity and heterogeneity of a sequence define the relationship of a symbol in a sequence with its next neighbor. Such relationships are neglected by parsimony-based distance measures. Parsimony is unable to define an evolutionary arrow of time: it cannot tell a parent from a daughter. Any two sequences are on equal footing. But here we find time dependent ordering and disordering, an arrow of time, and sometimes an ability, in a probabilistic sense, to decide which of two sequences is an evolutionary predecessor. Random mutation causes sequences to move toward random mix of symbols having neither perfect homogeneity nor perfect heterogeneity.

The Model

We use the model defined in the previous chapter. We consider an alphabet of k characters (a, b, c, ...). For example k = 4 nucleotides represents DNA or k = 20 amino acids represents proteins. We begin at time $t = 0$ with a parent sequence, labeled P_0. At each tick of an event clock, our computer algorithm inserts, deletes, and substitutes every possible character at every possible site to create an ensemble of daughter sequences. The first generation daughter sequences are then used as parent sequences to produce the next generation of daughters, t = 2, which then serve as the parent sequences for the next generation, t = 3, and so on. Unlike the weighted model of the previous chapter, multiple instances of a daughter sequence are not collected together and assigned a single statistical weight. Rather, we can now tabulate the number of daughter sequences with the same levels of degeneracy. This is compared to the weighting of the previous chapter.

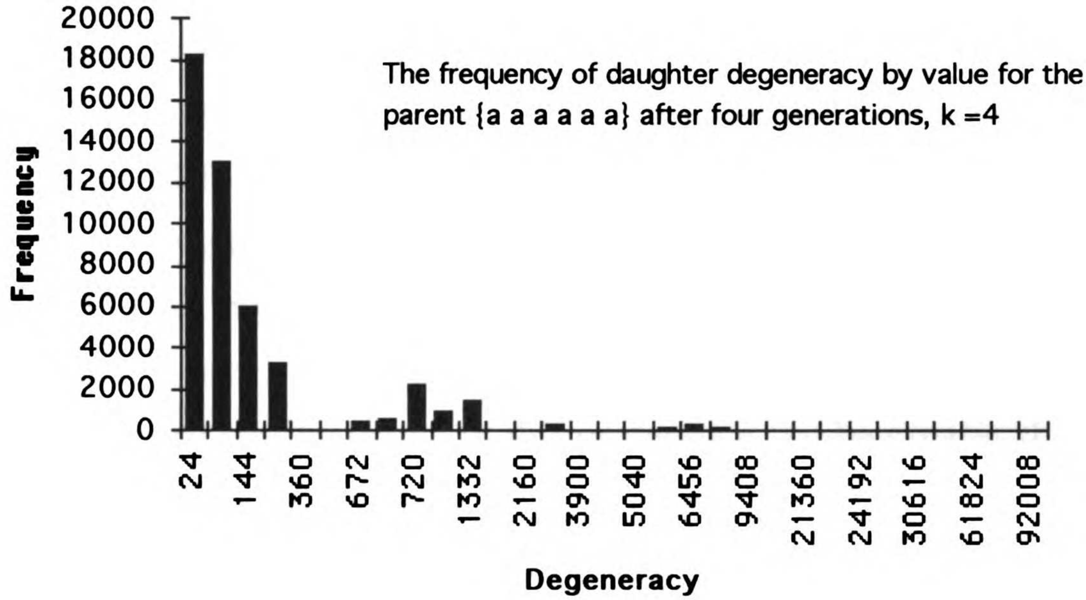
Results

Parent Sequence Composition and Daughter Sequence Degeneracy

Figure III-1 shows the distribution of the number of daughters of the given parent having a given level of degeneracy after two generations. It is clear that the homogeneous parent sequence {a a a a a a} has only 31 different levels of degeneracy after four generations. The amount of degeneracy ranges from 18225 daughter sequences having a 24 fold degeneracy to 1 daughter sequence having a 92008 fold degeneracy. The heterogeneous parent sequence {a b c d a b}, on the other hand, has 393 different levels of degeneracy, from 19875 daughters having a degeneracy of 24 to one daughter having a degeneracy of 40050. While there are more levels of degeneracy for the heterogeneous parent, they are less populated. Most degeneracy levels for the heterogeneous parent are populated by only two daughter sequences.

Both plots show that the majority of daughters have relatively small degeneracies after four generations. In analogy with the cities on a roadmap, it is equivalent to having few big cities and many small towns. This is because the number of paths (number of permutations of events) is independent of the distribution of characters in original parent sequences of the same length. It is interesting that the number of sequences at the lowest level of degeneracy (24 fold degenerate for either daughter space) is greater for the daughter space from the heterogeneous parent, it essentially has more "small towns".

A



B

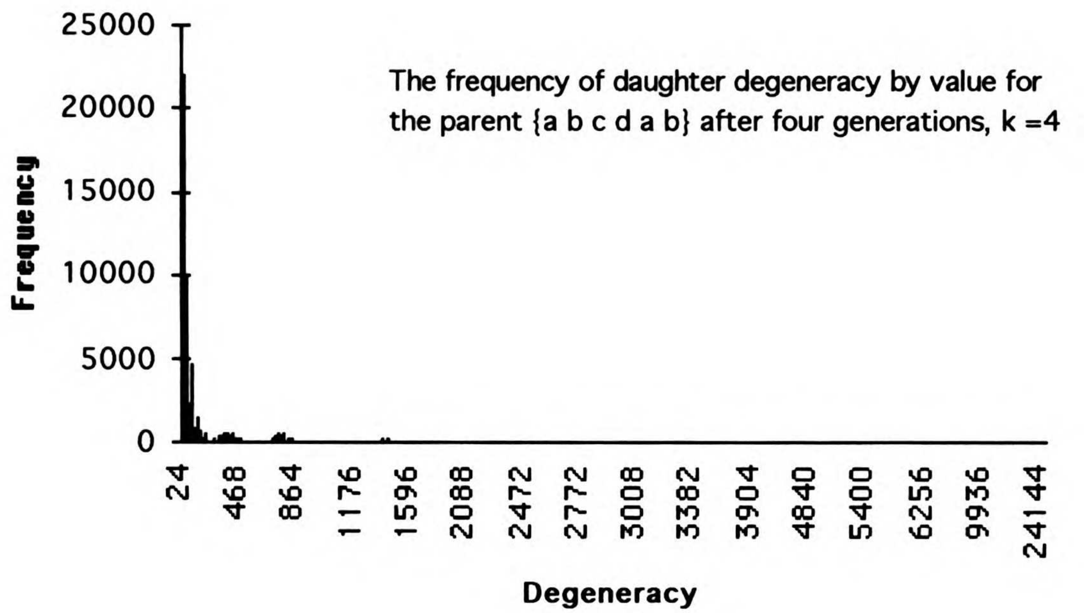
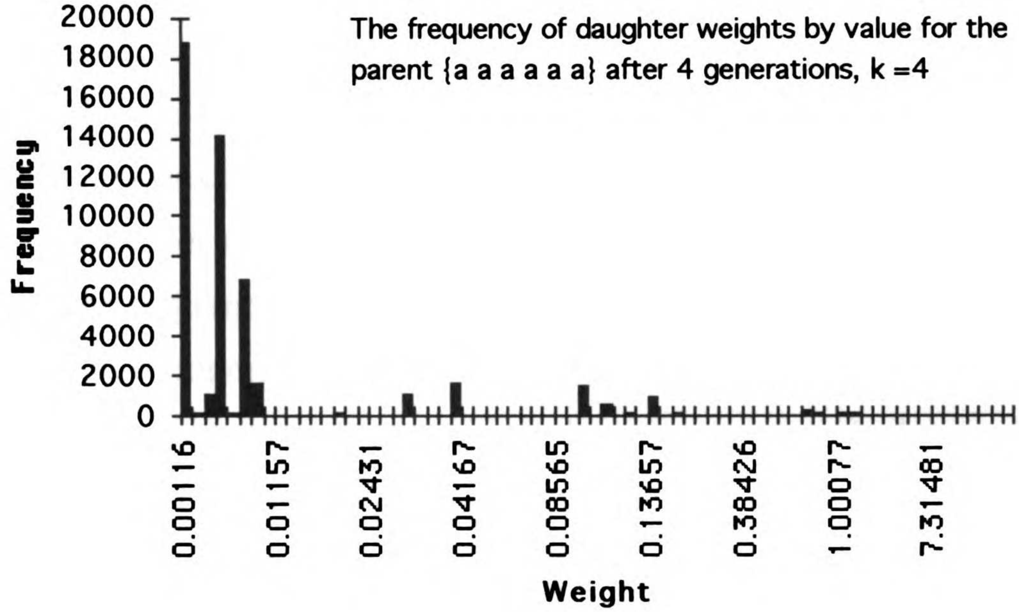


Figure III-1 A The number of daughter sequences by specific level of degeneracy for the parent sequence {a a a a a a} after four events. **B** The number of daughter sequences by specific level of degeneracy for the parent sequence {a b c d a b} after four events. Tick marks are absent for the columns, all 393 categories can not be labeled so X axis labels are for reference only.

The distribution of weights within homogeneous and heterogeneous daughter spaces is disparate after four generations (fig. III-2). While there is only one daughter sequence with the weight 33.15 for the homogeneous parent, this equals 153% of the combined weight of all 18711 of the sequences with a weight of 0.001157 (the most common weight) and 1.84% of the total weight of daughter space. Not only are some sequences favored in daughterspace but the extent of this preference can be greater than a hundred fold and the biggest "city" is larger than all of the smallest "towns" combined. The largest weight for a sequence in the heterogeneous daughter space is 10.62, only 24% of the combined weight of the most highly populated weight state, 19428 daughters weighing 0.002315, and only 0.59% of the total weight of daughter space. While this is comparatively less striking than the example of the homogeneous daughter space, it is important to note that without homogeneity there are still mechanisms for creating degeneracy and one sequence can be favored over another.

A



B

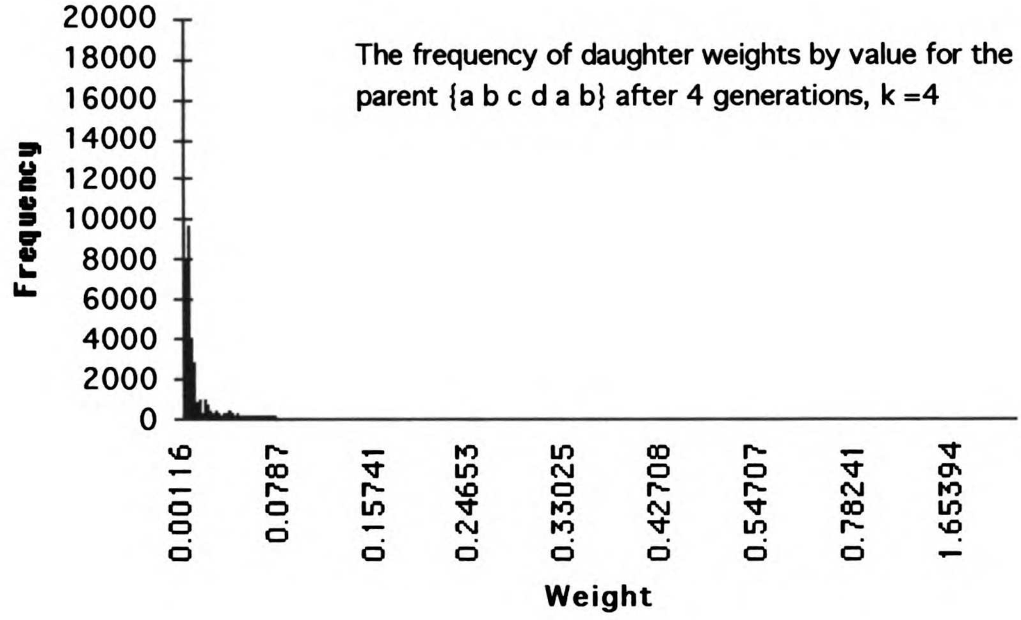


Figure III-2 A The number of daughter sequences by weight for the parent sequence {B:a a a a a a} after four events. All 71 categories can not be labeled so X axis labels are for reference only. **B** The number of daughter sequences by weight for the parent sequence {a b c d a b} after four events. Tick marks are absent for the columns, all 580 categories can not be labeled so X axis labels are for reference only.

The plots of weight and degeneracy decay more rapidly for heterogeneous sequences than for homogeneous sequences. This is not simply because there are more degeneracy values among which daughter sequences can be distributed: the maximum occupation of the lowest degeneracy and weight values are very similar. For heterogeneous sequences, there is a more rapid decay in the occupation of both the higher degeneracy and weight values. Among heterogeneous sequences, we observe degeneracy mostly due to the predisposition of a sequence to return to a state similar to the parent sequence. By analogy, random walking crosses most often through the origin regardless of the number of paths. For homogeneous sequences, degeneracy is increased by the ability of events in one part of a sequence to correct for those in another part.

The evolutionary processes of homogeneous sequences are different than for heterogeneous sequences. There are more heterogeneous than homogenous sequences to which a homogenous parent sequence can mutate. But a homogeneous pattern in a parent sequence has more degenerate paths to daughter sequences containing that same pattern or a permutation of that pattern. It is interesting to note that repetitive sequences such as a TATA box have a propensity to attain a daughter sequence similar to the parent sequence even when they are affected by mutations other than slipped strand mispairing (Levinson and Gutman 1987), a simple misalignment of DNA strands facilitated by the repetitive pattern of a TATA box. According to our model, homogeneous patterns are better conserved than heterogeneous sequences. The actual characters and sites may change but the patterns shared by these sites with their neighbors are more likely to re-emerge.

Characterizing Sequence Heterogeneity Ordering

Here we explore the time dependence of sequence ordering and disordering as a parent evolves through mutation. We define order parameters based on three different patterns:

1) Dyads: the same character appears in two adjacent positions in a sequence e.g. {v y **a a** z}.

2) Triads: the same character appears in three adjacent positions, e.g. {y **a a a** z} (**a a a** is also counted as two dyads).

3) Dyad Pairs: a pair of characters appear adjacent to the same pair of characters, e.g. {y **a b a b** z}.

The "ordering" of a daughter sequence D, $O_{d,\text{pattern}}(D)$, with respect to a given pattern is the number of occurrences of the pattern divided by the maximum number of times the repetitive pattern could possibly appear in that sequence:

$$O_{d,pattern}(D) = \frac{N_{pattern}}{l_s - l_o + 1} \tag{III-1.1}$$

where l_s is the length of sequence D, l_O is the length of the repetitive pattern and $N_{pattern}$ is the number of times the repetitive pattern appears in sequence D. Normalizing this by the total weight of all sequences in a daughter space, the order of a daughter space, $O_{g,pattern}(P_0, t)$, within each generation t is calculated as

$$O_{g,pattern}(P_0,t) = \frac{\sum_{D_n \in D_w(P_0,t)} W(D_n, D_w(P_0,t)) \bullet O_{d,pattern}(D_n)}{\sum_{D_n \in D_w(P_0,t)} W(D_n, D_w(P_0,t))} \tag{III-1.2}$$

where D_n is a member of $D_w(P_0,t)$

The expected amount of order for a random collection of sequences, $\langle O_{pattern} \rangle$, is dependent upon the number of repeated characters and the size of the character set:

$$\langle O_{pattern} \rangle = \prod_R \frac{1}{k^{(N_R - 1)}} \tag{III-1.3}$$

where R is the number of distinct characters which repeat, k is the character set size and N_R is the number of repetitions in the pattern.

Thus for k = 4, the expected value of $\langle O_{dyad} \rangle$ is 0.25 (with one run R = 1 and N_1 = 2). For a dyad pair, there are two repetitions (R = 2), both of two characters (N_1 and N_2 both equal 2), and the product must be taken over two character repitions making $<O_{g,\ pair}>$ = 0.0625. For a triplet R = 1, N_1 = 3 and $<O_{x,\ triplet}>$ = 0.0625. As the mutational clock ticks, $O_{g,pattern}$ tends toward its expectation value whether approached from higher homogeneity (figure III-3) or lower homogeneity (figure III-4).

This defines an arrow of time. A sequence having perfect homogeneity or perfect heterogeneity is more likely to be a parent sequence, and a more "randomly mixed" sequence is more likely to be a daughter. Given a family of related daughter sequences it is possible to determine if the sequences were derived from a parent with more or less order than the daughter sequences. If the sequences are above the steady-state value it is likely the ancestral sequence was more ordered. If the sequences are below the steady-state value it is likely the ancestral sequence was less ordered. When the ancestral sequence is extremely homogeneous or heterogeneous it provides a basis for estimating the relative clock rate of clades and OTUs from the relative degrees of ordering within a subset of the daughter sequences.

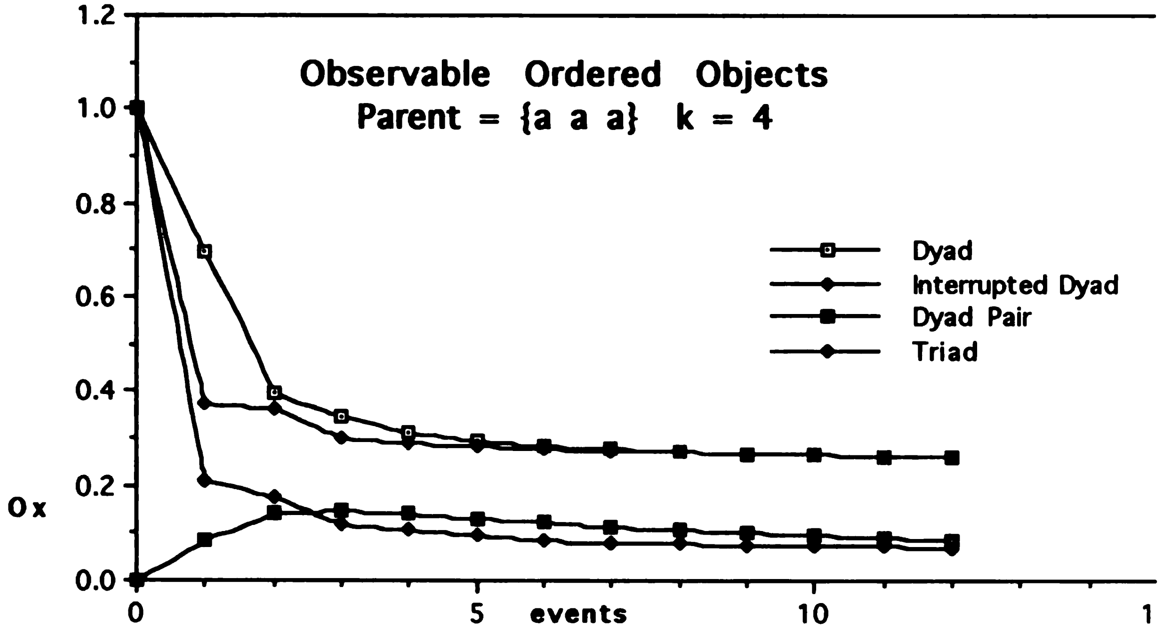


Figure III-3 Equilibrium values approached from the homogeneous parent sequence {C: a a a}.

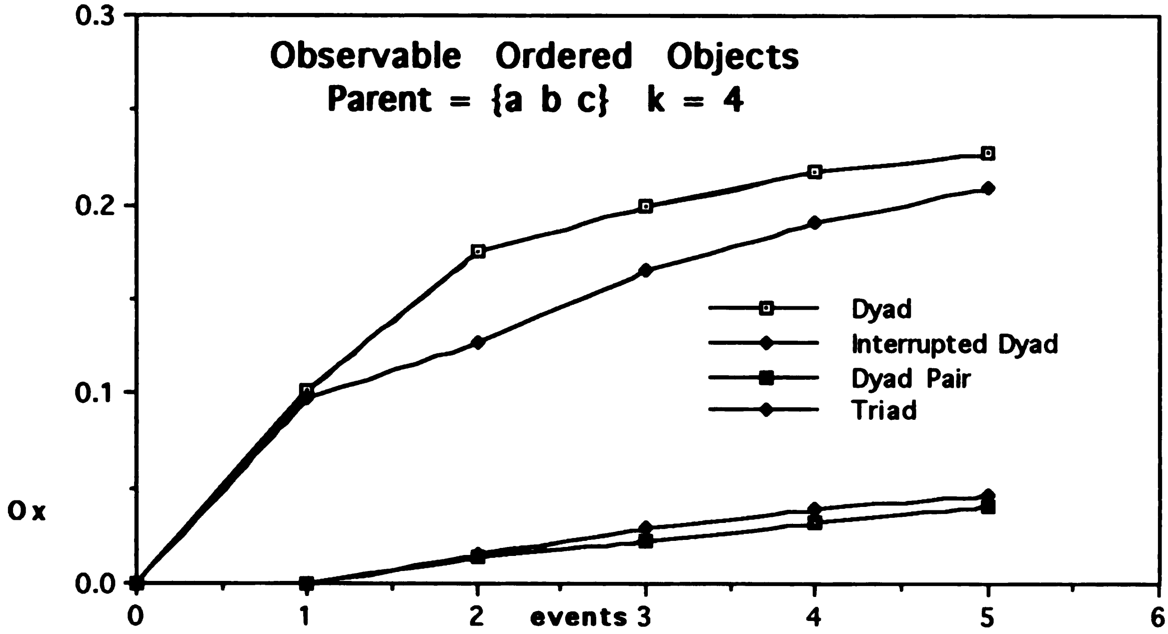


Figure III-4 Equilibrium values approached from the heterogeneous parent sequence {C: a b c}.

Conclusions

We have explored a simple computer model of the random evolution of sequences of symbols such as the monomers in biomolecules. It is not an exact model of biological evolution. There is no selection, for example. Rather, this is simply a model of principle to explore unbiased mutational processes from a parent to its daughter sequences, in part to assess parsimony-based distance measures. Here we have explored first the distribution of degeneracies of daughters from a given parent sequence. We find a sort of clustering - some sequences have high degeneracies (i.e. many mutational routes) and some have lower degeneracies (fewer mutational routes). Even when the same number of events are used the probability of realizing a more or less favored daughter sequence is determined not just by the probability of the events occurring but by the degeneracy of the daughter. These factors together determine the weight of a daughter sequence.

By defining relationships between neighboring characters in a sequence, we can establish order parameters that are useful for studying the time dependence of sequence evolution. We have explored the time dependence of ordering and disordering as homogeneous parent sequences evolve to broader compositions of symbol types and heterogeneous parent sequences evolve to more "randomly mixed" distributions of character types. An arrow of time is able to distinguish, in a probabilistic sense, which of a pair of sequences is more likely to be the parent and which the daughter.

Chapter 4

Testing End Effects in an Exact Enumeration Model of Sequence Mutation

Abstract

In previous chapters, we explored an exact enumeration model of mutational change in biomolecule sequences. In those chapters, we studied "bound" sequences, sequences that are assumed to be taken from a larger genome that provides a fixed set of "bookends" defining the beginning and end of the sequence. In this chapter, we ask whether our conclusions about limitations of parsimony-based distance measures are an artifact of end effects that might arise because of our assumption that the sequences in our simulations are bound by immutable positions next to both termini. Here we consider "unbound" sequences, which are unrestricted at the ends. Despite the added complexity of the calculation, we find the same problems with parsimony-based measures from this model of unbound sequences as we found in the model of bound sequences, implying that our conclusions about the limitations of parsimony are robust and independent of how end effects are treated.

Introduction

In this chapter we explore the sequence evolution model described in earlier chapters using different assumptions about how mutations can affect the ends of evolving sequences. In previous chapters sequences such as {a a a} were regarded as being taken from some larger genome, bound at the beginning and end by immutable characters, say x and y. This assumption has a consequence for defining whether a particular sequence is identified as a daughter sequence or not. Using italics to represent the immutable positions, {*x* a a a *y*} is not considered to match {*x* b a a a b *y*}. But if instead a sequence {a a a} is taken to be unbound, i.e., not contained between two immutable characters, then the appearance of a daughter sequence {b a a a b} can be said to contain {a a a} for the purpose of testing what daughters can be observed from a parent.

Therefore we define a "probe" sequence to be a detector ({a a a} in the case above) that is passed through a daughter space to establish statistical weights. A “match” is counted when a probe recognizes any sequence or subsequence in daughter space. We then pool the weights of all matching supersequences when calculating the probability of realizing the probe sequence as a daughter sequence. Bound sequences are denoted by a “B:” before the parent sequence, e.g. {B: a b a a}. Unbound sequences are denoted by a “C:” before the parent sequence, e.g. {C: a b c d}.

The Model

The model is the same as in previous chapters, with two differences. First, we must treat insertions differently at the ends. The total weight of all daughter sequences created by insertions is equal to the weight of all daughter sequences created by deletions when $f_i = f_d$. The probability of a substitution, insertion or deletion is a "per position per generation" quantity. This is maintained by treating insertions as if they occurred at an existing position in the sequence, just as deletions do.

Following Bishop and Thompson (1986), we take the insertion probability at either side of a position in the sequence to be half the probability of an insertion (per position), $f_i/2$. The probability of an insertion at a given site depends upon whether there are one or two characters adjacent to the site. To determine the contribution of a daughter sequence created by insertion to daughter space: if the site is internal it contributes f_i; if it flanks the sequence, adjacent to only one site, the weight is $f_i/2$. The weight of a sequence created by an insertion at a site flanking an unbound sequence is adjusted by $f_i/2$, since there is only one position adjacent to the insertion site. Therefore a weight factor, F, is included in the simulation of sequence mutation to reflect the diminished probability of flanking insertions for sites at the ends of unbound sequences (see example below).

Bound sequences are not modeled in their entirety. Two portions of the supersequence are removed from the sequence which is used in a simulation of mutation, one trimmed from each side of the bound sequence: there are conserved sequence positions adjacent to the insertion site flanking the bound sequence. The model for the mutation of bound sequences demonstrates a tendency for growth in the length of a daughter sequence. This should not be dismissed as artifactual under this model. The growth is permitted due to the fact that insertions at the site flanking the bound sequence simulation do not alter the trimmed sequences.

The second difference in the simulations in this chapter, compared to earlier chapters, is that here we delete sequences of zero length from daughter space. Sequences are not allowed to arise from nothing. In the simulations of bound sequences, because there are delimiters at the ends, insertions are permitted.

Example

Table IV-1 shows an example computer simulation of sequence mutation. D_w({C:a a c},1) is the first generation obtained from the parent {C:a a c}. Weights are obtained as follows.

1) Each substitution creates a new daughter with a statistical weight $w_d = w_p f_s/k$ where w_p is the weight of the immediate parent.

2) Each deletion creates a new daughter with a statistical weight $w_d = w_p f_d$.

3) Each insertion creates a new daughter with a statistical weight $w_d = w_p f_i F/k$ where F is the flanking factor (F = 1/2 for insertions at the flanking sites of unbound sequences, otherwise F = 1).

Substitution	w_d	Deletion	w_d	Insertion	w_d
{**a** a c}*	1/12	{a c}*	1/3	{**a** a a c}*	1/24
{**b** a c}	1/12	{a c}*	1/3	{**b** a a c}	1/24
{**c** a c}	1/12	{a a}	1/3	{**c** a a c}	1/24
{**d** a c}	1/12			{**d** a a c}	1/24
{a **a** c}*	1/12			{a **a** a c}*	1/12
{a **b** c}	1/12			{a **b** a c}	1/12
{a **c** c}	1/12			{a **c** a c}	1/12
{a **d** c}	1/12			{a **d** a c}	1/12
{a a **a**}	1/12			{a a **a** c}*	1/12
{a a **b**}	1/12			{a a **b** c}	1/12
{a a **c**}*	1/12			{a a **c** c}*	1/12
{a a **d**}	1/12			{a a **d** c}	1/12
				{a a c **a**}	1/24
				{a a c **b**}	1/24
				{a a c **c**}*	1/24
				{a a c **d**}	1/24

Table IV-1 **The immediate daughter sequences of P_0 = {C:a a c} with $f_s = f_d = f_i = 0.333$ and the character set (a, b, c, d): k = 4.. * denotes a degenerate sequence: it appears more than once in the table but once with a total weight in the weighted daughter space. Bold indicates the changed character for insertions and deletions. Below is the daughter space, and weights, for D_W({C: a b c}, 1).**

{a a c}*	1/4,	{b a c}	1/12,	{c a c}	1/12,
{d a c}	1/12,	{a b c}	1/12,	{a c c}	1/12,
{a d c}	1/12,	{a a a}	1/12,	{a a b}	1/12,
{a a d}	1/12,	{a c}*	2/3,	{a a}	1/12,

{a a a c}*	5/24,	{b a a c}	1/24,	{c a a c}	1/24,
{d a a c}	1/24,	{a b a c}	1/12,	{a c a c}	1/12,
{a d a c}	1/12,	{a a b c}	1/12,	{a a c c}	1/8,
{a a d c}	1/12,	{a a c a}	1/24,	{a a c b}	1/24,
{a a c d}	1/24.				

Results

Accessibilities

With this model of unbound sequences, we make the same tests as in earlier chapters for bound sequences. Figure IV-1 shows a case in which the parsimony-based shortest path correctly ranks the closeness of two parent daughter relationships. The relationship $D_1|P_1$ is {d a a a a b}|{C: b c d a b c} and the relationship $D_2|P_2$ is {d a a a a b}|{C: c d a a b c}. The latter parent/daughter relationship is rapidly ascending over the course of the simulation and will not converge to zero. In this computational sense, unbound sequences are more challenging than bound sequences.

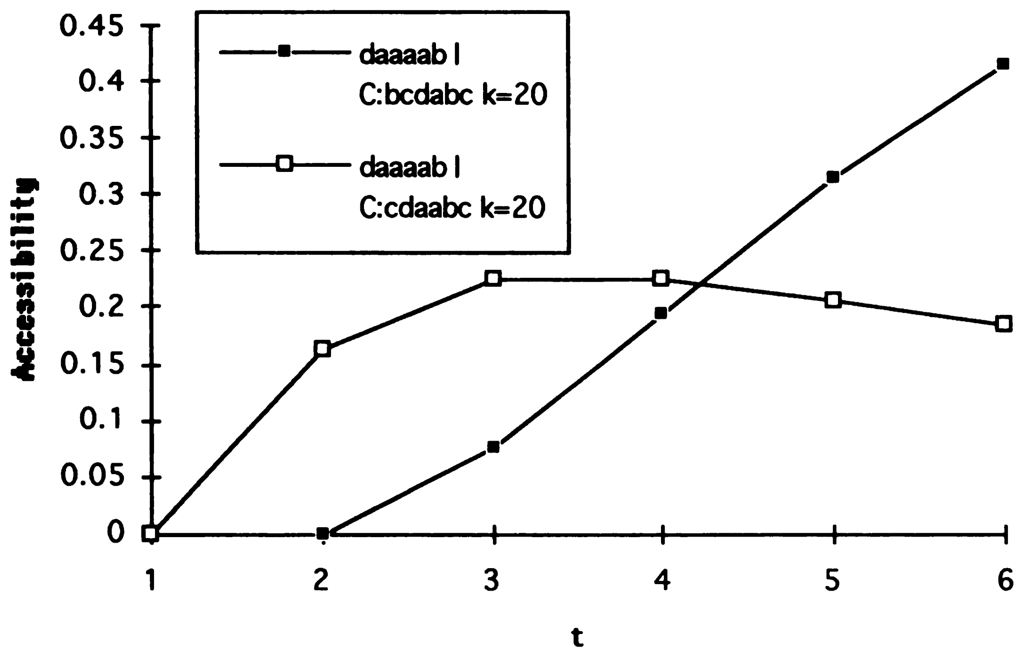


Figure IV-1 The accessibility of {d a a a a b}|{C: b c d a b c} and the accessibility of {d a a a a b}|{C: c d a a b c} as functions of generation time.

Figure IV-2 shows a problem with parsimony. Figure IV-2 corresponds to figure II-1, where we considered two cases in which the daughter is identical to its parent. D_1 is {a a a a a a}, a daughter of parent P_1: {B: a a a a a a}, and D_2 is {a b c d a b}, a daughter of parent P_2: {B: a b c d a b}. In this simple case, parsimony defines both parent/daughter relationships as being equally close, since both have Hamming distances equal to zero. In contrast, it is clear from the figure that the accessibilities (closeness of the relationships) are different.

In early generations there is less difference between the accessibilities of the unbound homogeneous and heterogeneous relationships when compared to figure II-1 for bound sequences. Because the probe sequence recognizes subsequences, the unbound homogeneous relationship accessibility profile is smoother than the corresponding profile for the bound relationship. An insertion of the appropriate character at any site in the unbound sequence or the insertion of any character adjacent to the unbound sequence does not require a balancing deletion.

This smoothing effect is best illustrated using high insertion and deletion probabilities. The bound parent/daughter relationship in figure IV-3 has large oscillations in its accessibility. The majority of pathways to reach the given daughter from a parent require paired mutations, an insertion and deletion. The unbound parent/daughter relationship can make use of pathways where insertions outnumber deletions.

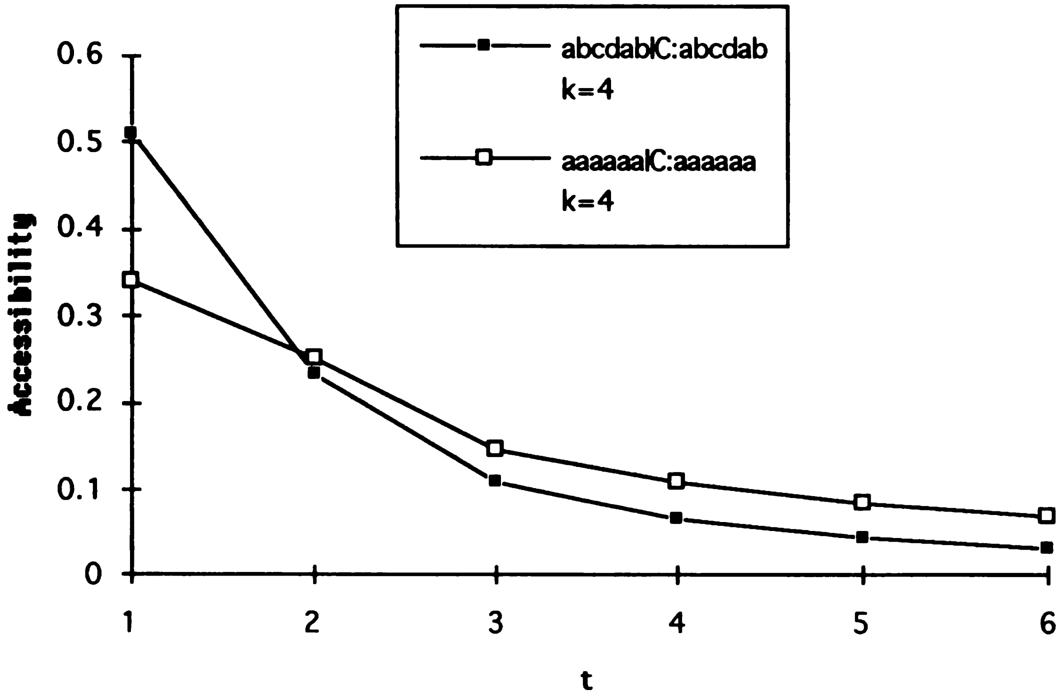


Figure IV-2 The accessibility of {a b c d a b}|{C: a b c d a b) as a function of time and the accessibility of {a a a a a a}|{C: a a a a a a} as a function of time.

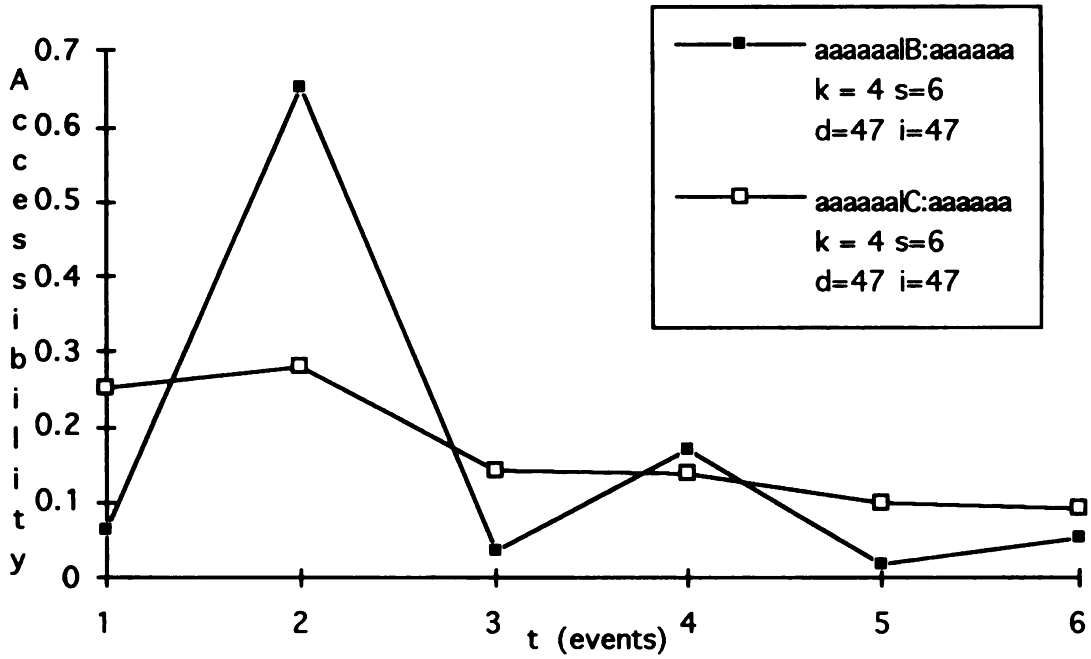


Figure IV-3 The accessibility of {a a a a a a}|B: a a a a a a} and {a a a a a a}|C: a a a a a a} as a function of time with $f_s = 0.06$ $f_d = 0.47$ $f_i = 0.47$.

Figure IV-4 tests the parsimony-based Levenshtein distance, as a generalization of the Hamming distance test in figure IV-2. These parent/daughter relationships are companions to the tests shown in and figure II-2 for bound sequences. Here too, the Levenshtein distance would rank these two parent/daughter pairs as equally close, but figure IV-4 shows they have very different accessibilities. There is a significant difference from the bound accessibility in the ability to realize the sequence {a a a} as a daughter of {C:a a a a a} because the daughter can be recognized as a subsequence of a daughter space sequence. It is clear that the unbound relationship is closer than the bound relationship. One must question, however, the value of analyzing parent/daughter relationships where the daughter sequence can be immediately recognized as a subsequence of the parent sequence without mutation; because none of the insertions are internal the Hamming distance could be used on aligned sequences.

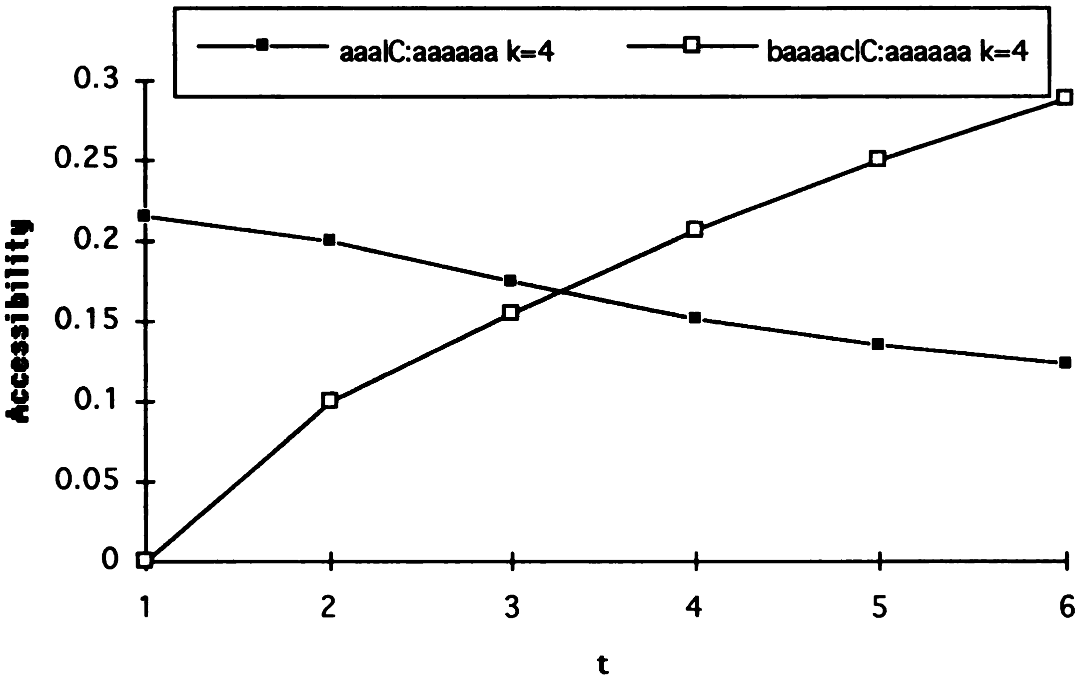


Figure IV-4 The accessibilities of {a a a}|{C: a a a a a a) and {b a a a a c}|{C: a a a a a a) as functions of time.

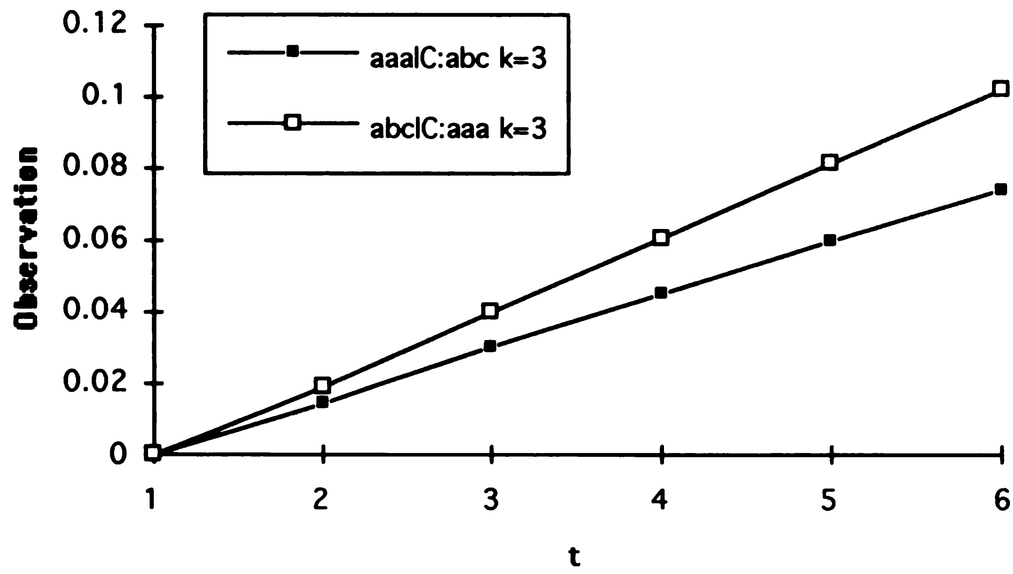
End Effects for Unbound Sequences

End effects can cause asymmetries in parent/daughter relationships. In figure IV-5 the probability of observing {a b c}|{C: a a a} is higher than that of observing {a a a}|{C: a b c} due to the insertion probabilities at flanking positions. After two events, the doubly inserted daughters {a **b** c a a}, {a a **b** c a} and {a a a **b** c} of the parent sequence

{C: a a a} match the probe sequence {a b c}. Similarly, for the parent sequence {C: a b c}, the doubly inserted daughter sequences {**a** **a** a b c}, {**a** a **a** b c} and {a **a** **a** b c} match the probe sequence {a a a}. However, the first set of matching daughter sequences has only one instance where flanking insertions play a role ({a a a **b** c}) while the second set of matching daughter sequences has two instances where flanking insertions play a role ({**a** **a** a b c} and {**a** a **a** b c}). The weights of daughter sequences having flanking insertions are half those of daughter sequences having internal insertions. Therefore the total weight of all the paths connecting the former parent/daughter relationship is greater than the weight of all paths connecting the latter relationship.

Also, when there are many paths to a given daughter sequence that involve events occurring within an "ordered object" (a pattern of similar characters in either the parent sequence or daughter sequence) the daughter arises more frequently than when there are fewer paths involving events occurring within an ordered object. Characters from the parent sequence that are irrelevant for matching the probe sequence can be shifted aside in mutational paths. There are more permutations available, and thus more pathways, using irrelevant characters than when all characters play a role in realizing a daughter sequence (see table IV-2).

A



B

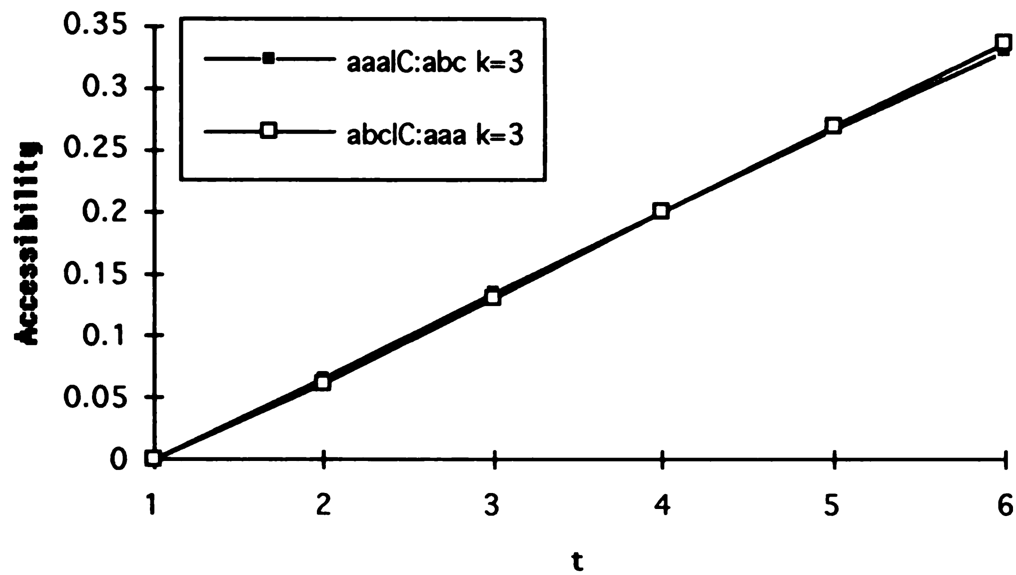


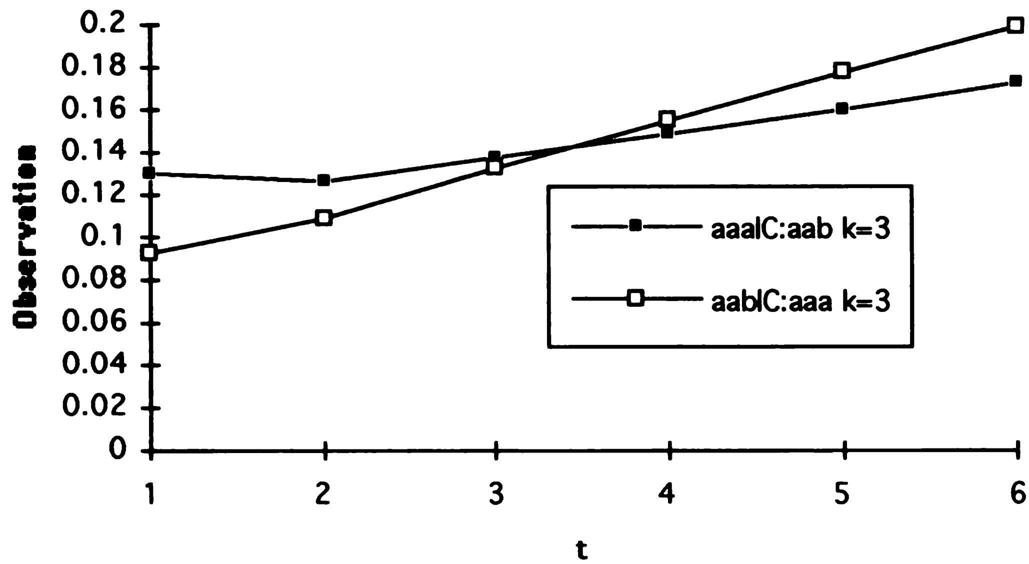
Figure IV-5 **A** The sequence {a b c} is more likely to be seen as a daughter of {C: a a a} than the sequence {a a a} as a daughter of {C a b c}. **B** The parent/daughter relationship {a b c}|{C: a a a} is less closely related than the parent/daughter relationship {a a a}|{C a b c}.

{a a a}|{C: a b a}:	{a **b** a} -> {a **a** a}
{a b a}|{C: a a a}:	{a **a** a} -> {a **b** a}; {a a a} -> {a **b** a *a*}; {a a a} -> {*a* a **b** a}
The heterogeneous sequence is favored as a daughter sequence after one generation	
{a a a}|{C: a a b}:	{a a **b**} -> {a a **a**}; {a a b} -> {a a **a** *b*}; {a a b} -> {a **a** a *b*};
	{a a b} -> {**a** a a *b*}
{a a b}|{C: a a a}:	{a a **a**} -> {a a **b**}; {a a a} -> {*a* a a **b**}
The homogeneous sequence is favored as a daughter sequence after one generation	

Table IV-2 Pathways for realizing a daughter sequence using one event. Surfeit characters are italicized.

Figures IV-6 and IV-7 show the asymmetries in parent/daughter accessibilities for unbound sequences.

A



B

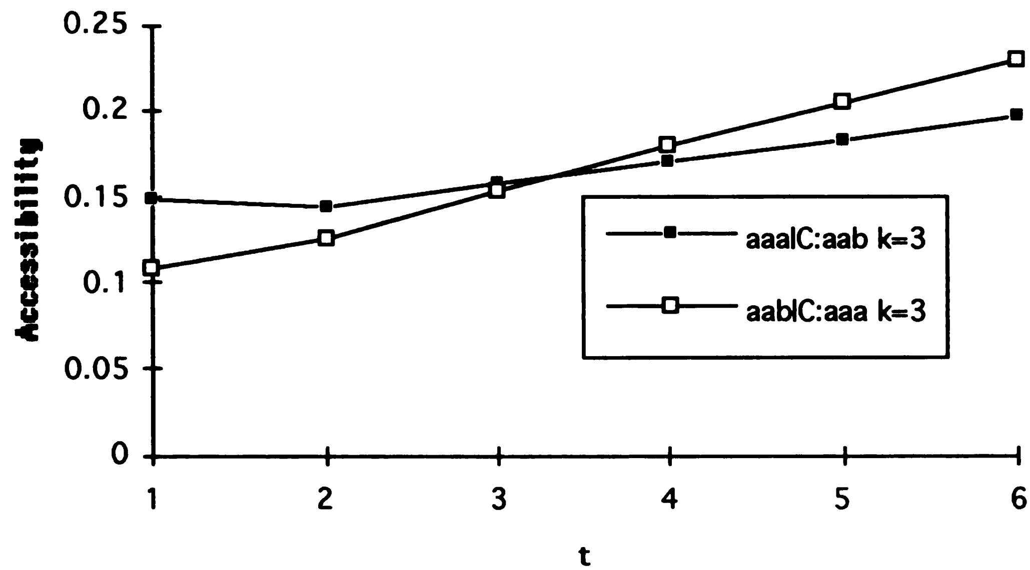
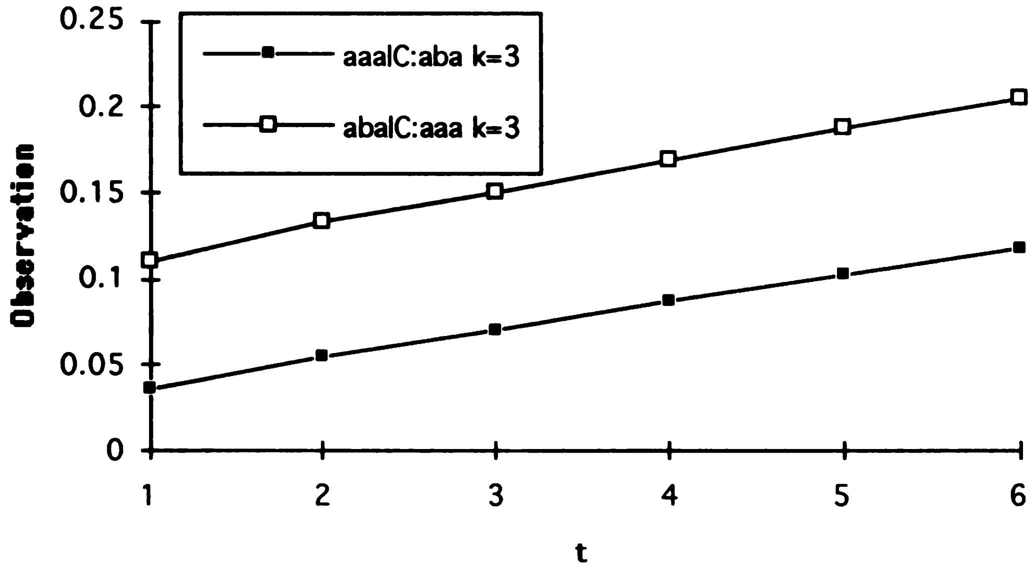


Fig. IV-6 **A** The relationship {a a a}|{a a b} is more likely to be seen in early generations than the relationship {a a b}|{a a a}. **B** The accessibility of relationships {a a a}|{a a b} and {a a b}|{a a a}.

A



B

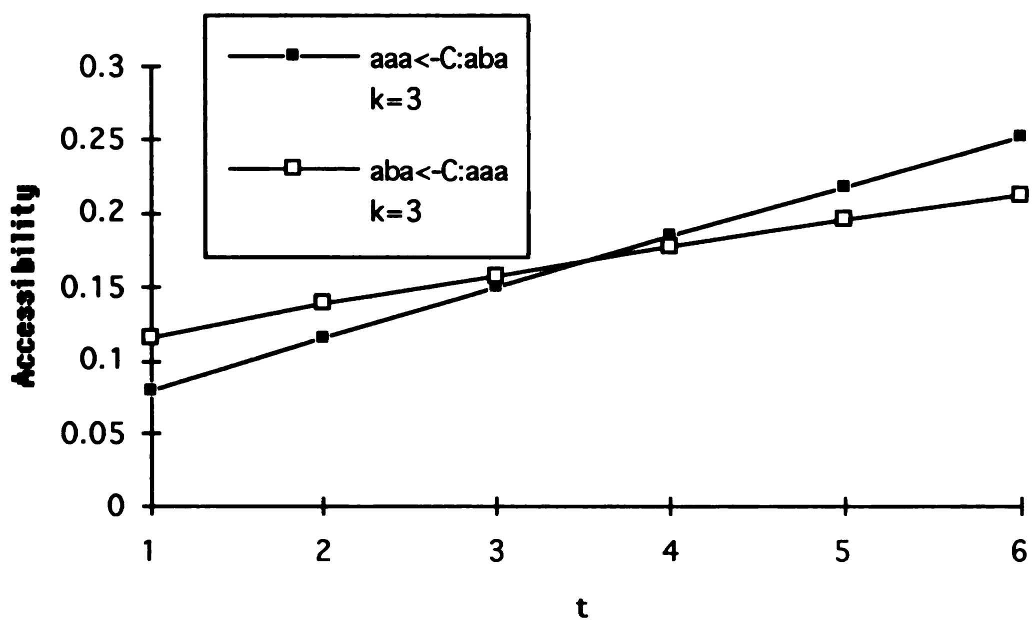


Fig. IV-7 **A** The relationship {a b a}|{a a a} is more likely to be seen in early generations than the relationship {a a a}|{a b a}. **B** The accessibility of relationships {a b a}|{a a a} and {a a a}|{a b a}.

Conclusions

Earlier conclusions made about parsimony are robust with respect to two different models of sequence mutation. In earlier chapters, we considered bound sequences, which are assumed to be bracketed by immutable bookend characters within a larger sequence. In this chapter we have considered unbound sequences to test whether our earlier conclusions about parsimony are altered. Although this model of unbound sequences is more unwieldy insofar as the computations are less convergent, nevertheless this model leads to the same general conclusions, namely that parsimony errs as a measure of sequence distance in its neglect of the multiple mutational pathways from a parent to a daughter.

Chapter 5

Convergence of the Kinetic Accessibility Property

Abstract

In previous chapters, we have studied the kinetic accessibilities of daughter sequences from parents through mutational cycles of evolutionary events. The kinetic accessibility is a quantity involving a normalization over time, t, i.e. mutational event cycles. Because mutational events occur even as t approaches ∞, we are interested in the question of whether the kinetic accessibility is convergent or divergent in the limit as $t \to \infty$. We analyze the asymptotic nature of a simple model of accessibility. We find that probabilities of observing a daughter sequence and kinetic accessibilities are generally convergent quantities in this model.

Introduction

In earlier chapters, we studied the generation of daughter sequences from parent sequences in a model of exhaustive mutational change. All the possible daughters of a parent in the first generation beget all the possible second generation daughters, which then beget all the possible third generation daughters, et cetera. If t defines the generation number, then this process can proceed indefinitely, $t \to \infty$. In this chapter we explore whether certain properties converge or diverge in the limit $t \to \infty$. In particular, we raise the questions;

1) Does the probability of observing a particular bound sequence approach zero?

That is, does $p(D_X, D_W(P_0,t)) \rightarrow 0$ as $t \rightarrow \infty$?

2) Does the expected distance converge?

That is, how does $t \bullet p(D_X,D_W(P_0,t))$ behave as $t \rightarrow \infty$?

3) Does the probability of **not** seeing a particular unbound sequence approach zero?

Specifically, does $1 - p(D_X,D_W(P_0,t)) \rightarrow 0$ as $t \rightarrow \infty$?

4) How does its expected value,

$t \bullet (1 - p(D_X,D_W(P_0,t)))$, behave as $t \rightarrow \infty$?

The Convergence of Sequence Lengths

Does the probability of a given daughter sequence converge as $t \rightarrow \infty$? Let us consider two cases. First consider no insertions or deletions, $f_i = f_d = 0$. Then it is clear that as $t \rightarrow \infty$ $p(D_X,D_W(P_0,t))$ converges to a constant, $(1/k)^n$, where k is the alphabet size and n the sequence length. Second, consider any non-zero insertion and deletion rate. Over time, the daughter space sequences will grow or shrink in length if there is an imbalance of insertions and deletion. If the daughter space sequences tend to grow, then as $t \rightarrow \infty$ the average length of daughter space sequences will also approach infinity and the probability of observing a finite length daughter approaches zero.

The perfect balance of growth and shrinkage does not occur at $f_i = f_d$ ($f_s \neq 1$) because of two factors that promote an increase in the mean length of daughter space sequences. The first is the "absorbing boundary condition", i.e. zero length sequences disappear from daughter space. This creates an imbalance by constantly removing heavily deleted sequences while heavily inserted sequences may grow without limit. The second factor is the sequence propagation effect. The probability of an event occurring takes place on a per position basis. Immediate parent sequences make a contribution to the immediate daughter space proportional to the number of positions they contain.

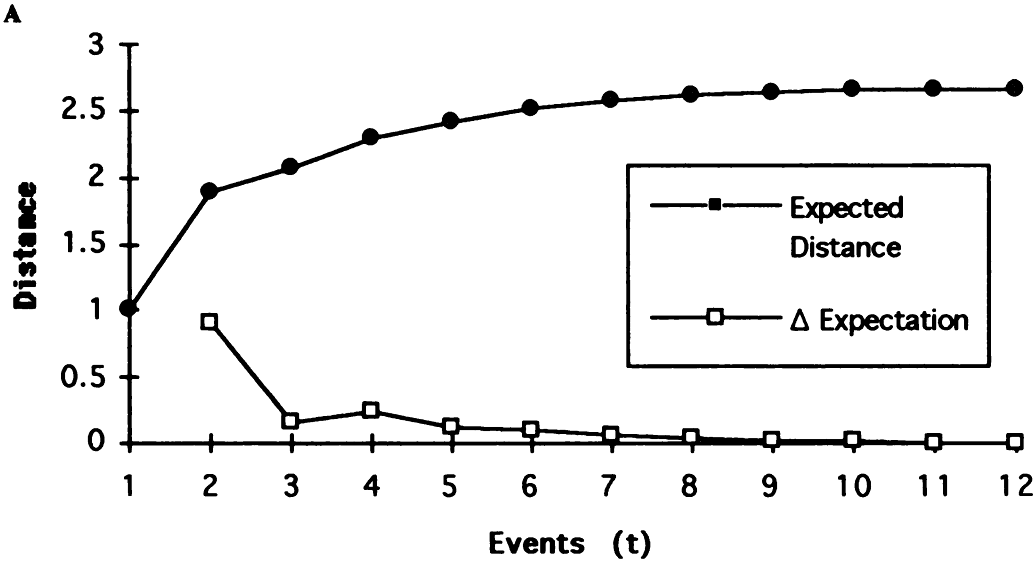
A most interesting situation arises that for any given daughter sequence length there will be some ratio of deletions to insertions, f_d/f_i, above which daughter sequences grow and below which daughter sequences shrink. High deletion probabilities will not cause daughterspace to disappear, probabilities of observation will simply be calculated using the daughter space that is extant. At precisely that one value of f_d/f_i, $p(D_X,D_w(P_0,t))$ converges to some finite value, not to zero; daughter space will balance the effects of absorption, which emphasize longer length sequences, with the effects of high deletion probabilities, which emphasize shorter length sequences.

Convergence of Distance in Sequences

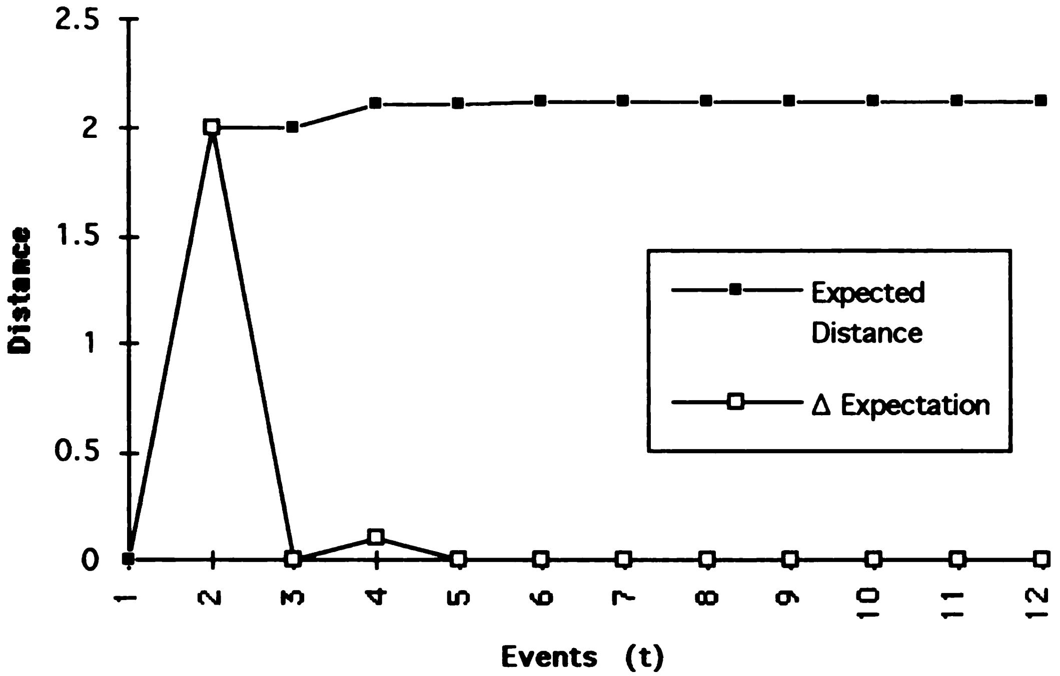
In order to study convergence over the largest possible number of generations on the computer, we consider the shortest possible sequence {a} as a daughter of {B:a}, in a character set of size k = 2.

Figure V-1 shows the expected distance between parent and daughter versus *t*. Figure V-1A shows the case of $f_s = f_d = f_i = 1/3$, where the distance converges (slowly), as indicated by the reduction in differences shown on the figure. Two asymmetric cases, $f_d = 0.1$, $f_i = 0.9$ and $f_d = 0.9$, $f_i = 0.1$ are also shown in figures V-1B and V-1C. The higher the insertion probability, the more rapid the convergence.

A



B



C

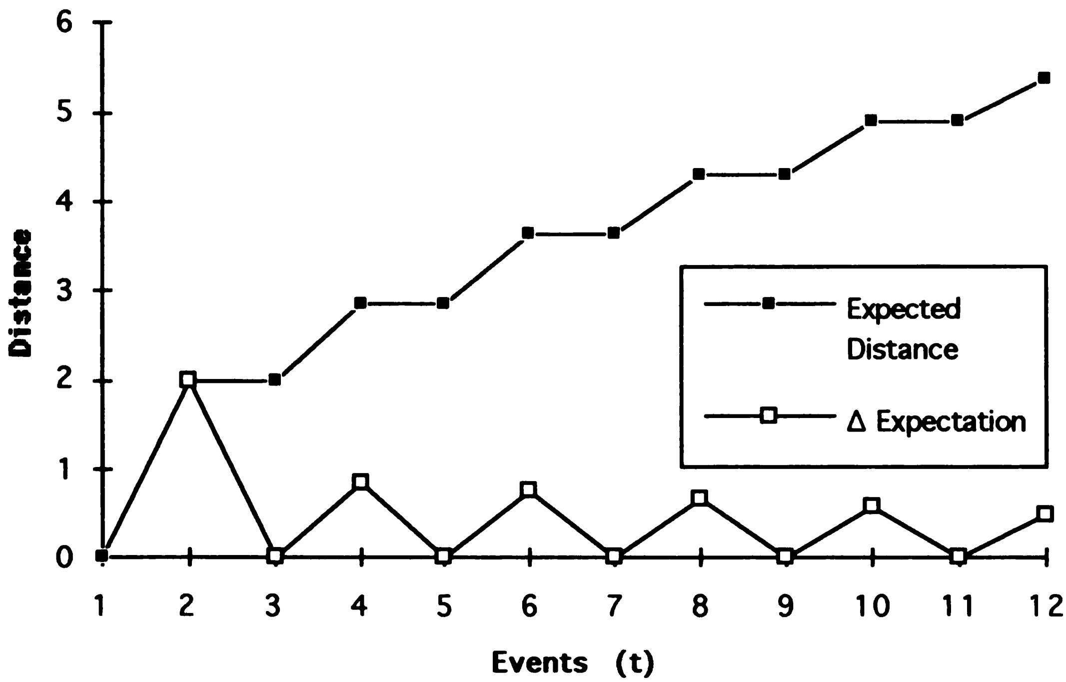


Figure V-1 A The expected distance of {a}|{B:a} as more events *t* are included in the simulation. $f_s = f_d = f_i = 0.33$. The lower curve plots the change in the expected observation between each event and the event before it. B The same plot with $f_d = 0.1, f_i = 0.9$. C The same plot with $f_d = 0.9, f_i = 0.1$.

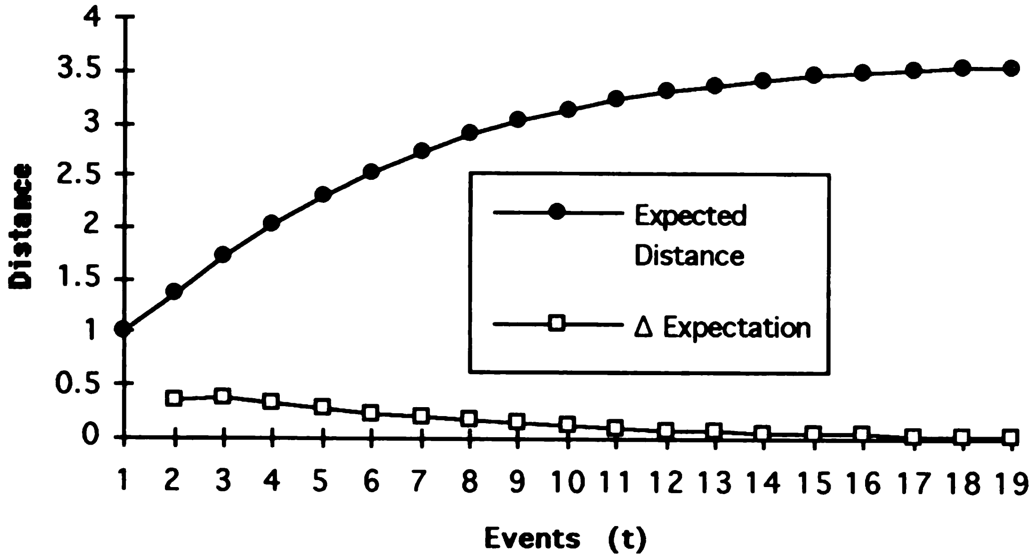
Unbound Sequences

There is a large qualitative difference between bound and unbound sequences. For unbound sequences, we use a daughter sequence of finite length to search a daughter space of growing sequence lengths. As t approaches infinity, the mean length of daughter sequences approaches infinity, so the probability approaches one that the probe will match *some subsequence* of the daughters. In this case, what should converge to zero is the probability that the probe fails to find a match, $1-p(D_X,D_w(P_0,t))$. The expected distance is now

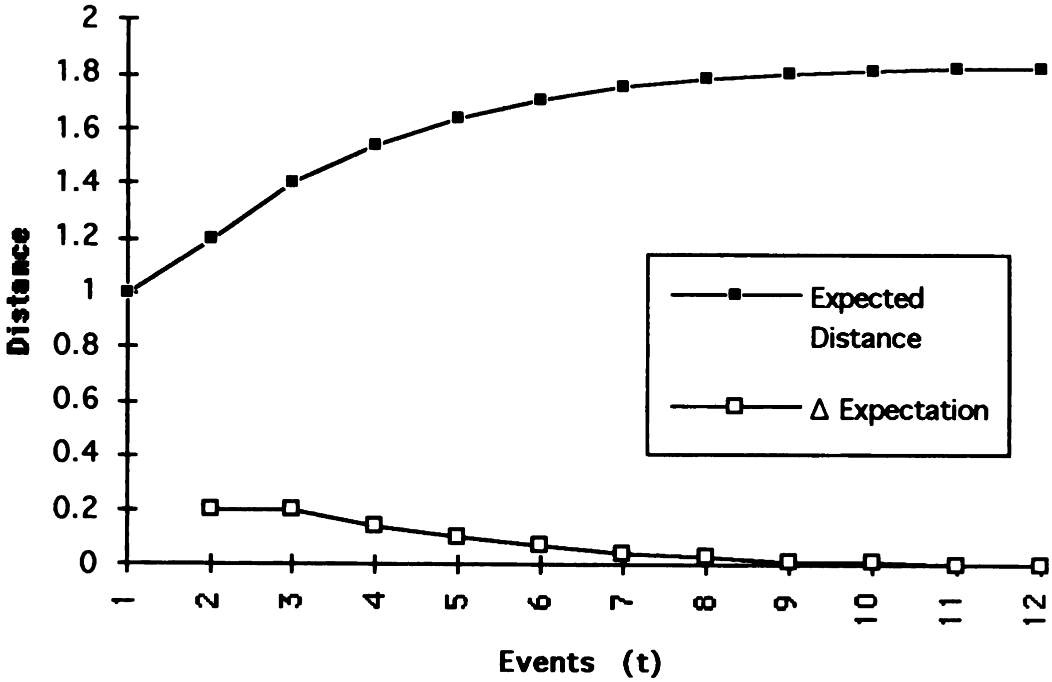
$$\langle t(D_X|P_0)\rangle = \frac{\sum_t t \bullet n(D_X,D_w(P_0,t))}{\sum_t n(D_X,D_w(P_0,t))} \tag{V-1}$$

Figure V-2 shows the three examples used above, but now for unbound sequences, and shows that these values converge as expected.

A


Distance
Expected Distance
Δ Expectation
Events (t)

B


Distance
Expected Distance
Δ Expectation
Events (t)

C

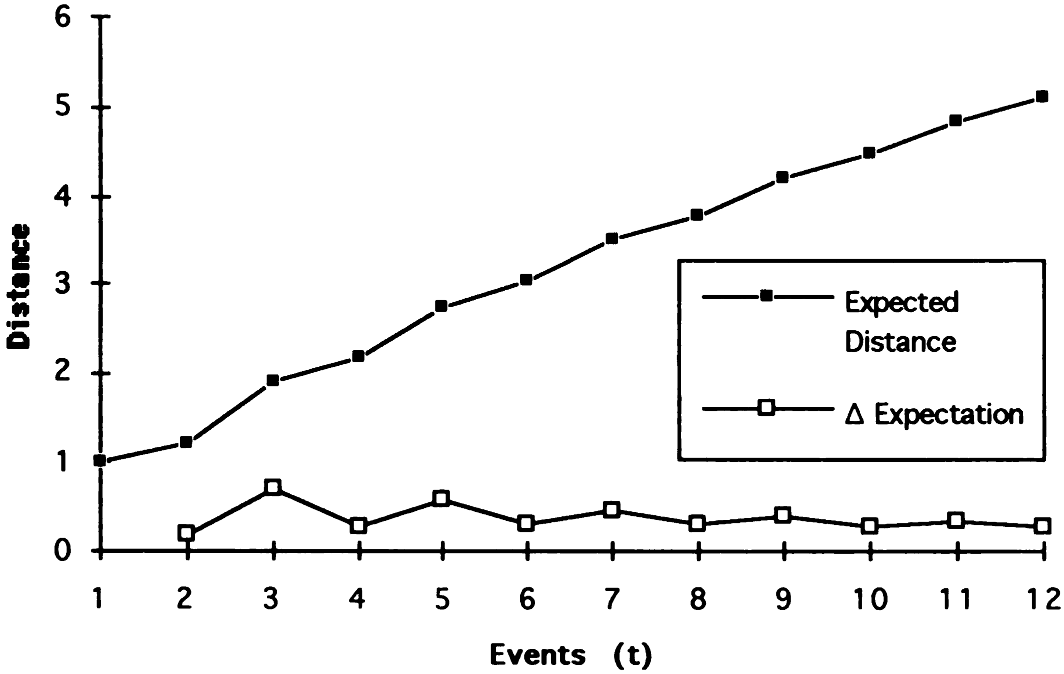


Figure V-2 **A** The expected distance for nonobservation {a}|{B:a} as more events t are included in the simulation. $f_s = f_d = f_i = 0.33$. The lower curve plots the change in the expected observation between each event and the event before it. **B** The same plot with $E_d = 0.1, E_i = 0.9$. **C** The same plot with $f_d = 0.9$, $f_i = 0.1$.

Conclusions

We have studied the convergence of properties of daughter sequences in the asymptotic limit of large numbers of mutational events *t*. We find that for non-zero insertion and deletion rates, the probability of a observing a given daughter approaches zero since the daughter space becomes biased against sequences of the appropriate matching length. For unbound sequences, this requires calculating the probability of not observing the probe sequence and the expected distance of not making an observation when the mean length of daughter space sequences approaches infinity. There will be a particular value of f_d/f_i at which there is a perfect balance of insertions and deletions, where the mean length of daughter space sequences will not vary and the accessibility does not decrease rapidly enough to attain convergence.

References:

Atchley WR, Fitch WM (1991) Gene trees and the origins of inbred strains of mice. Science 254:554-558

Ayala FJ, Kiger JA (1980) Modern Genetics. Benjamin/Cummings Publishing Company, Menlo Park pp 304-306

Bishop MJ, Thompson EA (1986) Maximum likelihood alignment of DNA sequences. J Mol Biol 190:159-165

Bull JJ, Cunningham CW, Molineux IJ, Badgett MR, Hillis DM (1993) Experimental molecular evolution of bacteriophage T7. Evolution 47:993-1007

Camin JH, Sokal RR (1965) A method for deducing branching sequences in phylogeny. Evolution 19:311-326

Cavalli-Sforza LL, Edwards AW (1967) Phylogenetic analysis: models and estimation procedures. Evolution 21:550-570

Cavendar (1978) Taxonomy with confidence. Math Biosci 40:271-280

DeBry RW (1992) The consistency of several phylogeny-inference methods under varying evolutionary rates. Mol Biol Evol 9:537-551

Eck RV, Dayhoff MO (1966) Atlas of protein sequence and structure. National Biomedical Research Foundation, Silver Springs pp 164-168

Farris JS (1972) Estimating phylogenetic trees from distance matrices. Am Nat 106:645-668

Felsenstein J (1978) Cases in which parsimony or compatibility methods will be positively misleading. Syst Zool 27:401-410

Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17:368-376

Felsenstein J (1988) Phylogenies from molecular sequences: inference and reliability. Ann Rev Genet 22:521-565

Fitch WM (1971) Toward Defining the course of evolution: minimum change for a specific tree topology. Syst Zool 20:406-416

Fitch WM, Margoliash E (1967) Construction of phylogenetic trees. Science 155:279-284

Gojobori T, Ishii K and Nei M (1982) Estimation of average number of nucleotide substitutions when the rate of substitution varies with nucleotide. J Mol Evol 18:414

Goldman, N (1990) Maximum Likelihood Inference of phylogenetic trees, with special references to a poisson process model of DNA substitution and to parsimony analysis. Syst Zool 39:345-361

Hasegawa M, Fujiwara M (1993) Relative efficiencies of the maximum likelihood, maximum parsimony, and neighbor-joining methods for estimating protein phylogeny. Mol Phylo Evol 2:1-5

Hendy MD, Penny D (1989) A framework for the quantitative study of evolutionary trees. Syst Zool 38: 297-309

Hillis DM, Bull JJ (1991) Of genes and genomes. Science 254:528-529

Hillis DM, Bull JJ, White ME, Badgett MR, Molineaux IJ (1992) Experimental phylogenetics: generation of a known phylogeny. Science 255:589-591

Hillis DM, Bull JJ, White ME, Badgett MR, Molineaux IJ (1993) Experimental approaches to phylogenetic analysis. Syst Biol 42:90-92

Hillis DM, Huelsenbeck JP, Cunningham CW (1994) Application and Accuracy of Molecular Phylogenies. Science 264:671-677

Hillis DM, Swofford DL (1994) Hobgoblin of phylogenetics? Nature 369:363-364

Huelsenbeck JP and Hillis DM (1993) Success of phylogenetic methods in the four-taxon case. Syst Biol 42:247-264

Jukes TH, Cantor CR (1969) Evolution of protein molecules. In: Munro HN (ed) Mammalian protein metabolism. Academic Press, NY pp 21-132

Kimura M (1980) A simple method for estimating evolutionary rate of base substitution through comparative studies of nucleotide sequences. J Mol Evol 16:111-120

Kimura M (1981) Estimation of evolutionary distances between homologous nucleotide sequences. PNAS 78:454-458

Kimura M (1983) The Neutral Theory of Molecular Evolution. Cambridge University Press, Cambridge

Kruskal JB (1983) An overview of sequence comparison. In: Kruskal JB, Sankoff D (ed) Time Warps, String Edits, and Macromolecules: the theory and practice of sequence comparison. Addison-Wesley Publishing Company, Reading pp 1-44

Kruskal JB, Sankoff D (ed) (1983) Time Warps, String Edits, and Macromolecules: the theory and practice of sequence comparison. Addison-Wesley Publishing Company, Reading pp 1-44

Levinson G, Gutman GA (1987) Slipped strand mispairing: a major mechanism for DNA sequence evolution. Mol Biol Evol 4:203-221

Lockhart PJ, Steel MA, Hendy MD, Penny D (1994) Recovering evolutionary trees under a more realistic model of sequence evolution. Mol Biol Evol 11:605-612

Maddison WP, Maddison DR (1992) MacClade: analysis of phylogeny and character evolution. Sinauer, Sunderland

Miyata T, Yasanuga T (1980) Molecular evolution of mRNA: a method for estimating evolutionary rates of synonymous and amino acid substitutions from homologous nucleotide sequences and its application. J Mol Evol 16:23-36

Nei M, Gojobori T (1986) Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. Mol Biol Evol 3:418-426

Palumbi SR (1989) Rates of molecular evolution and the fraction of nucleotide positions free to vary. J Mol Evol 29:180-187

Perler F, Efstratiadus A, Lomedico P, Gilbert W, Kolodner R, Dodgson J (1980) The evolution of genes: the chicken preproinsulin gene. Cell 20:555

Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Bio Evol 4:406-425

Sankoff, D (1975) Minimal mutation trees of sequences. SIAM J Appl Math 28:35-42

Sidow A (1994) Parsimony or statistics? Nature 367:26

Shoemaker JS, Fitch WM (1989) Evidence from Nuclear Sequences that invariable sites should be considered when sequence divergence is calculated. Mol Biol Evol 6:270-289

Sneath PHA, Sokal RR (1973) Numerical Taxonomy. WH Freeman and Company, San Francisco pp 230-234

Sober E (1993) Experimental tests of phylogenetic inference- methods. Syst Biol 42:85-89

Stewart CB (1993) The powers and pitfalls of parsimony. Nature 361:603-607

Swofford DL, Olsen GJ (1990) In: Hillis DM, Moritz C (ed) Molecular systematics. Sinauer, Sunderland pp 411-501

Swofford DL (1992) PAUP: Phylogenetic Analysis Using Parsimony, version 3.0s (Software and Manuals distributed by the Center of Biodiversity , Illinois Natural History Survey, Champaign, IL 61820)

Takahata N, Kimura M (1981) A model of evolutionary base substitutions and its application with special reference to rapid change of pseudogenes. Genetics 98:641-657

Tateno Y (1990) A method for molecular phylogeny construction by direct use of nucleotide sequence data. J Mol Evol 30:85-93

Tateno Y, Nei M, Tajima F (1982) Accuracy of estimated phylogenetic trees from molecular data. J Mol Evol 18:837-404

Thorne JL, Kishino H, Felsenstein J (1992) Inching toward Reality: An Improved Likelihood Model of Sequence Evolution. J Mol Evol 34:3-16

Wagner WH (1961) Problems in the classification of ferns. In: Recent advances in botany. 841-844 University of Toronto Press, Toronto

Yang, Ziheng (1996) Phylogenetic analysis using parsimony and likelihood methods. J Mol Evol 42:294-307

Appendix

Facilitation of Computation

Methods

This appendix describes the computer algorithm we used for the sequence generation work in my thesis. The principle of the algorithm is as follows. The original parent sequence P_0 is given. It comprises the "daughter" space $D_w(P_0,0)$ which is used to generate the daughter sequences which comprise $D_w(P_0,1)$. The collection of all possible immediate daughter sequences created from $D_w(P_0,1)$ are treated as immediate parent sequences to form the set $D_w(P_0,2)$. Each subsequent generation is recursively dependent on the generation before it. This is how the algorithm works in theory. But in practice it is computationally intensive to implement directly. We use four tricks to improve performance.

1) Condensation

The first trick is a condensation step. Degenerate daughter sequences are removed (except for one instance) from the daughter space as each generation is formed, even if the generation is an intermediate generation in a simulation proceeding to generations of greater t. The final operation when creating any generation of $D_w(P_0,t)$ is to pool the weight of each instance of a degenerate daughter sequence in the calculated daughter space and assign that weight to the one instance of the daughter sequence present in $D_w(P_0,t)$, e.g. the daughter sequence {a a c c} of the parent {C:a a c} appears twice in Table IV-1 (with the weights 0.04166 and 0.08333) but is present once in $D_w(P_0,1)$ (with the weight 0.125). The condensation step reduces i) the number of sequences used to calculate $D_w(P_0, t+1)$, ii) the corresponding calculation time for $D_w(P_0, t+1)$, and iii) the amount of computer storage required by $D_w(P_0, t)$. Under most circumstances condensation is desirable; calculation of all $D_w(P_0,n)$ for $0 < n < t$ is performed so savings due to condensation are compounded in later generations. Some information is lost. The actual count of degeneracy is no longer known for each sequence and any information about a pathway that could be inferred from the weight of one instance of the degenerate sequence is forfeited.

2) Wildcards

The second trick is the introduction of “wildcards” for insertions and substitutions. A wildcard is the character "?" introduced into a sequence as the replacement for a substituted character or an inserted character. The wildcard is a placeholder symbol that stands for all possible characters in the character set. A sequence containing a wildcard represents all k possible daughter sequences that are created by a substitution or insertion. A ramification of this is that a true $D_w(P_0,t)$ is not created. Rather, we create a database containing wildcard sequences. A wildcard sequence carries the weight of all sequences it represents. Another program is used to extract specific daughter sequences from the database by i) finding all sequences which match the probe sequence (the wildcard matching any character) and ii) adjusting the weight of the matching daughter sequences for the character set size. Under the assumption of equiprobability of characters, the weight of a database sequence is distributed evenly among all sequences possible when all the wildcards are expanded to all possible characters. This saves computer time in two ways: i) the same database calculations for a given parent may be reused for any applicable alphabet and ii) a substitution or deletion that falls on a previous insertion or substitution may represent all k paths between the two wildcard generating events. An applicable alphabet must have a character set size greater than the number of discrete characters in the original parent sequence; it is not possible to use an alphabet where k = 3 if the original parent sequence has four different characters (e.g. {a b c d}).

3) Pattern Databases

The third trick is that of the creation of a pattern database. A pattern sequence may be used to represent all original parent sequences having a length equal to that of the pattern sequence used to create the pattern database. The pattern sequence uses placeholder symbols for every position in the original parent sequence. If there are 10 positions in the original pattern sequence then 10 different placeholder symbols are used to hold each of the positions. Once the pattern database is created the daughter space for any specific original parent sequence of the same length may be extrapolated from this database. Using a mapping program, the characters in each position of a given P_0 are mapped to the placeholder symbols in the pattern daughter space. This produces an intermediate database of daughter sequences containing wildcards. Given that the mapping of characters from P_0 may reveal degenerate sequences which can be condensed, the wildcards may be treated as characters with respect to a condensation step. Then a program for wildcard expansion is run and the list of daughter sequences without wildcards is condensed a second time yielding $D_w(P_0,t)$ for the given P_0 with a character set size k.

The calculation time is longer for pattern databases because fewer degeneracies can be removed during the condensation steps between the calculation of each generation: each position is a unique symbol and no overlap of symbols from different positions, other than wildcards, can occur until the pattern is replaced with the characters of an original parent sequence. The pattern database is larger than a regular wildcard database for the same reason. It also takes more time to extrapolate a parent/daughter relationship from daughter space using the pattern database; much of the workload for the calculations needed to analyze sequence properties has been "back ended" from the up front cost of building a database to the analysis stage after the database has been created. This is compensated for by i) the decrease in storage space due to there being only one database for all P_0's of a given length and ii) an overall calculation time decrease for multiple P_0's that can make use of the pattern database as a starting point for building their own weighted daughter spaces.

Condensation steps made before a pattern is replaced with a given parent sequence reveal that there are two main classes of degeneracies, "event" and "character" degeneracies. Degeneracies that occur in pattern databases prior to the replacement of the pattern with characters from a parent sequence are those created by the order of events. Event degeneracies do not depend on the characters in a sequence: the creation of a position followed by a later deletion of the same position or the deletion of two positions in any order will create degenerate daughters regardless of the characters at those positions. These event degeneracies can be observed in the pattern database before the mapping step used in data extrapolation. Character degeneracies, on the other hand, cannot be observed until a pattern has been replaced with characters. These can be further subdivided into two classes, "compositional" and "insertion" degeneracies. Compositional degeneracies occur when a discrete character is present at more than one position in a sequence, e.g. {**a** a a b} -> {a a b} and {a a **a** b} -> {a a b} shows multiple instances of the same degenerate daughter created by one deletion (using the form "->" to indicate a step across one event). Insertion degeneracies are an inherent property of insertions, {a *x*} -> {a ***x*** *x*} and {a *x*} -> {a *x* ***x***} show that an insertion degeneracy must occur if insertions are allowed no matter what character is used for *x*. An insertion may be considered an event that may force a local homogeneity regardless of the homogeneity or heterogeneity of the parent sequence.

4) Event Databases

The fourth trick is an event database that allows one database to be used for all f_s, f_d, and f_i. Three variables called event counters are associated with each daughter sequence pattern. These correspond to the number of substitutions, deletions and insertions. The number of each type of event is counted but not used in the calculation of weights while the database of sequences is generated. This creates a database whose weights are influenced only by the aforementioned event degeneracies and flanking insertion factors because i) the character set size is not introduced until wildcards are expanded, ii) character degeneracies are not introduced until the pattern has been replaced with characters from an original parent sequence and iii) event weights are not introduced until the database is queried for daughter sequences using a specific set of weights. Condensation steps made before event weights are implemented can only pool sequences with the same event counter values. The added flexibility of a database with event counters decreases calculation time overall and decreases storage space if and only if calculations will be performed for the same parent/daughter relationship using varying substitution, deletion and insertion probabilities.