# 2

# Words

## 2.1 The Biological Problem

Consider the DNA sequence shown below:

```
TGATGATGAAGACATCAGCATTGAAGGGCTGATGGAACACATCCCGGGGCCGGAC
TTCCCGACGGCGGCAATCATTAACGGTCGTCGCGGTATTGAAGAAGCTTACCGTA
CCGGTCGCGGCAAGGTGTATATCCGCGCTCGCGCAGAAGTGGAAGTTGACGCCAA
AACCGGTCGTGAAACCATTATCGTCCACGAAATTCCGTATCAGGTAAACAAAGCG
CGCCTGATCGAGAAGATTGCGGAACTGGTAAAAGAAAAACGCGTGGAAGGCATCA
GCGCGCTGCGTGACGAGTCTGACAAAGACGGTATGCGCATCGTGATTGAAGTGAA
ACGCGATGCGGTCGGTGAAGTTGTGCTCAACAACCTCTACTCCCAGACCCAGTTG
CAGGTTTCTTTCGGTATCAACATGGTGGCATTGCACCATGGTCAGCCGAAGATCA
TGAACCTGAAAGACATCATCGCGGCGTTTGTTCGTCACCGCCGTGAAGTGGTGAC
CCGTCGTACTATTTTCGAACTGCGTAAAGCTCGCGATCGTGCTCATATCCTTGAA
GCATTAGCCGTGGCGCTGGCGAACATCGACCCGATCATCGAACTGATCCGTCATG
CGCCGACGCCTGCAGAAGCGAAAACTGCGCTGGTTGCTAATCCGTGGCAGCTGGG
CAACGTTGCCGCGATGCTCGAACGTGCTGGCGACGATGCTGCGCGTCCGGAATGG
CTGGAGCCAGAGTTCGGCGTGCGTGATGGTCTGTACTACCTGACCGAACAGCAAG
CTCAGGCGATTCTGGATCTGCGTTTGCAGAAACTGACCGGTCTTGAGCACGAAAA
ACTGCTCGACGAATACAAAGAGCTGCTGGATCAGATCGCGGAACTGTTGCGTATT
CTTGGTAGCGCCGATCGTCTGATGGAAGTGATCCGTGAAGAGCTGGAGCTGGTTC
GTGAACAGTTCGGTGACAAACGTCGTACTGAAATCACCGCCAACAGCGCAGACAT
CAACCTGGAAGATCTGATCACCCAGGAAGATGTGGTCGTGACGCTCTCTCACCAG
GGCTACGTTAAGTATCAGCCGCTTTCTGAATACGAAGCGCAGCGTCGTGGCGGGA
```

Given this sequence, there are a number of questions we might ask:

- What sort of statistics should be used to describe this sequence?
- Can we determine what sort of organism this sequence came from based on sequence content?
- Do parameters describing this sequence differ from those describing bulk DNA in that organism?

– What sort of sequence might this be: Protein coding? Centromere? Telomere? Transposable element? Control sequence?

This chapter approaches these sorts of questions from three different perspectives, all of which are united by considering **words**. These are short strings of letters drawn from an alphabet, which in the case of DNA is the set of letters A, C, G, and T. A word of length $k$ is called a "$k$-word" or "**$k$-tuple**"; we use these terms interchangeably. For example, individual bases are 1-tuples, dinucleotides are 2-tuples, and codons are triplets, or 3-tuples. GGGT is a 4-word or 4-tuple.

DNA sequences from different sources or regions of a genome may be distinguished from each other by their $k$-tuple content. We begin by giving illustrations and examples of how word frequencies can vary within and between genomes. Second, we take a closer look at computational issues pertaining to words, starting with the seemingly obvious one about how to count words and how words can be located along a string. This includes describing their distribution in terms of a probabilistic model. Finally, we discuss various statistics that have been used to describe word frequencies.

## 2.2 Biological Words: $k = 1$ (Base Composition)

We consider first the frequencies of individual bases. For free-living organisms (in contrast with some bacteriophages and other viruses), DNA is typically duplex. This means that every A on one strand is matched by a T on the complementary strand, and every G on one strand is matched by C on the complementary strand. In other words, the number of A residues in the genome equals the number of T residues, and the number of G residues equals the number of C residues. This applies to the duplex DNA: as we will see later, the number of G or A residues on one strand need not equal the number of C or T residues (respectively) on the same strand. This statement is illustrated below:

| | |
|---|---|
| 5′-GGATCGAAGCTAAGGGCT-3′ | Top strand: 7 G, 3 C |
| 3′-CCTAGCTTCGATTCCCGA-5′ | Duplex molecule: 10 G, 10 C |

Considering G or C for this particular *duplex* DNA, it suffices to report the fraction fr(G+C) of G+C, knowing the individual base frequencies will be fr(G+C)/2. Also, fr(A+T) = 1 − fr(G+C), so only a single parameter is required to describe the base frequencies for duplex DNA. (That is, there are four variables, fr(A), fr(C), fr(G), and fr(T), and there are three relations among them, fr(A) = fr(T), fr(G) = fr(C), and fr(A+T) = 1 − fr(G+C).)

Since the early days of molecular biology (before cloning and DNA sequencing), base composition has been used as a descriptive statistic for genomes of various organisms. The fraction fr(G+C) of bulk DNA can be determined either by measuring the melting temperature of the DNA, $T_m$,

or by determining the buoyant density of the DNA in a CsCl gradient using equilibrium ultracentrifugation. Both methods can reveal the presence of genome fractions having different base compositions, and in the case of ultracentrifuge measurements, bands of genomic DNA adjacent to the main band and differing from it in base composition may be described as "satellites."

Table 2.1 presents the base compositions of DNA from a few organisms, indicating the range over which this statistic can vary. Obviously, there are constraints on base composition imposed by the genetic code: long homopolymers such as $\cdots$ AAAAA $\cdots$ (fr(G+C) = 0) would not encode proteins having biochemical activities required for life. We see more about this when we consider $k = 3$ (codons).

**Table 2.1.** Base composition of various organisms. Bacterial data taken from the Comprehensive Microbial Resource maintained by TIGR: `http://www.tigr.org/tigr-scripts/CMR2/CMRHomePage.spl`.

| Organism | %G+C | Genome size (Mb) |
|---|---|---|
| Eubacteria | | |
| *Mycoplasma genitalium* | 31.6 | 0.585 |
| *Escherichia coli* K-12 | 50.7 | 4.693 |
| *Pseudomonas aeruginosa* PAO1 | 66.4 | 6.264 |
| Archaebacteria | | |
| *Pyrococcus abyssi* | 44.6 | 1.765 |
| *Thermoplasma volcanium* | 39.9 | 1.585 |
| Eukaryotes | | |
| *Caenorhabditis elegans* (a nematode) | 36 | 97 |
| *Arabidopsis thaliana* (a flowering plant) | 35 | 125 |
| *Homo sapiens* (a bipedal tetrapod) | 41 | 3,080 |

Another point will be visited in a later chapter: the distribution of individual bases within the DNA molecule is not ordinarily uniform. In prokaryotic genomes in particular, there is an excess of G over C on the **leading strands** (strands whose $5'$ to $3'$ direction corresponds to the direction of replication fork movement). This can be described by the "GC skew," which is defined as $(\#\mathtt{G} - \#\mathtt{C})/(\#\mathtt{G} + \#\mathtt{C})$, calculated at successive positions along the DNA for intervals of specified width ("windows"); here, #G denotes the number of Gs and so on. As will be seen later, this quantity often changes sign at positions of replication origins and termini in prokaryotic genomic sequences. This is

another example of how a relatively simple statistic based on $k$-tuples with $k = 1$ can be informative.

In the sections that follow, we develop some probabilistic and statistical approaches for describing the base composition, dinucleotide composition, and other aspects of DNA sequences. To do this, it is most convenient to describe the DNA in terms of a single strand in a given $5'$ to $3'$ orientation. The other strand of the duplex is formed by taking its complement.

## 2.3 Introduction to Probability

This is a good point to introduce some necessary basic ideas of probability. We build on these ideas as we progress through this and later chapters. In this section, we use the nucleotide sequence on a single strand as an example. For definiteness, we assume that this sequence is written in a given $5'$ to $3'$ direction. We are often interested in deciding whether particular patterns of bases appear unusually often in a given sequence; such sequences might be of biological significance. To address such problems, we need a way to measure our "surprise" about the frequency of particular patterns, and to do this we use a probabilistic model for the sequence.

One way to specify such a probability model is to describe a method for simulating observations from the model. This means that we must specify the probabilistic rules the computer uses to produce the next letter in the simulated sequence, given the previous letters. We can then think of the sequence as the output of a machine (or simulation algorithm). Here is a simple set of rules that specify a probability model:

(a) The first base in the sequence is either an `A`, `C`, `G`, or `T` with probability $p_{\mathtt{A}}, p_{\mathtt{C}}, p_{\mathtt{G}}, p_{\mathtt{T}}$, respectively.
(b) Suppose the machine has generated the first $r$ bases. To generate the base at position $r + 1$, the machine pays no attention to what has been generated before and spits out `A`, `C`, `G`, or `T` with the probabilities given in (a) above.

A run of the simulation algorithm results in a sequence of bases, and different runs will typically result in different sequences. The output of a random string of $n$ bases will be denoted by $L_1, L_2, \ldots, L_n$, where $L_i$ denotes the base inserted in position $i$ of the sequence. It is conventional to use small letters to denote the *particular* sequence that resulted from a run; we may observe $L_1 = l_1, L_2 = l_2, \ldots, L_n = l_n$ for a particular simulation. In the next sections, we outline some basic probabilistic and statistical language that allows us to analyze such sequences.

### 2.3.1 Probability Distributions

Suppose that on a single step our machine produces an output $X$ that takes exactly one of the $J$ possible values in a set $\mathcal{X} = \{x_1, x_2, \ldots, x_J\}$. In the DNA

sequence example, we have $J = 4$ and $\mathcal{X} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. We do not typically know with certainty which value in $\mathcal{X}$ will be produced by our machine, so we call $X$ a discrete **random variable**. (Note the font used to distinguish bases from random variables.) The term *discrete* refers to the fact that the set of possible values is finite. Now suppose that the value $x_j$ occurs with probability $p_j$, $j = 1, 2, \ldots, J$. We note that each $p_j$ must be greater than or equal to 0, and the $p_j$ must sum to 1; that is,

$$\begin{aligned} &p_1, p_2, \ldots, p_J \geq 0; \\ &p_1 + p_2 + \cdots + p_J = 1. \end{aligned}$$

We call the collection $p_1, \ldots, p_J$ the **probability distribution** of $X$, and we write

$$\mathbb{P}(X = x_j) = p_j, j = 1, 2, \ldots, J.$$

In this book, we always use the symbol $\mathbb{P}$ to denote probability. For example, the first base $L_1$ in our model for a DNA sequence has probability distribution

$$\mathbb{P}(L_1 = \mathtt{A}) = p_{\mathtt{A}}, \mathbb{P}(L_1 = \mathtt{C}) = p_{\mathtt{C}}, \mathbb{P}(L_1 = \mathtt{G}) = p_{\mathtt{G}}, \mathbb{P}(L_1 = \mathtt{T}) = p_{\mathtt{T}}. \tag{2.1}$$

Note that some textbooks use the term *probability mass function* of the random variable instead of *probability distribution*, defined above. The probability distribution allows us to compute probabilities of different outcomes in the following way. If $S$ is an event (that is, a subset of $\mathcal{X}$), then the probability that $S$ occurs, written $\mathbb{P}(X \in S)$, is calculated as

$$\mathbb{P}(X \in S) = \sum_{j : x_j \in S} p_j.$$

The term $j : x_j \in S$ is read "$j$ such that $x_j$ is in $S$." For example, if $S = \{\mathtt{G}, \mathtt{C}\}$, then $\mathbb{P}(X \in S) = p_{\mathtt{G}} + p_{\mathtt{C}}$.

In the following sections, we study the probability distribution of the number of times a given pattern occurs in a random DNA sequence $L_1, L_2, \ldots, L_n$, and we'll make our patterns one base long to begin with. To address this question, we define a new sequence $X_1, X_2, \ldots, X_n$ by

$$X_i = \begin{cases} 1, \text{ if } L_i = \mathtt{A}, \\ 0, \text{ otherwise.} \end{cases} \tag{2.2}$$

The number of times $N$ that $\mathtt{A}$ appears is then the sum of the $X$s:

$$N = X_1 + X_2 + \cdots + X_n. \tag{2.3}$$

Starting from the probability distribution (2.1) of the $L_i$, we can find the probability distribution of each of the $X_i$ as follows:

$$\begin{aligned} &\mathbb{P}(X_i = 1) = \mathbb{P}(L_i = \mathtt{A}) = p_{\mathtt{A}}, \\ &\mathbb{P}(X_i = 0) = \mathbb{P}(L_i = \mathtt{C} \text{ or } \mathtt{G} \text{ or } \mathtt{T}) = p_{\mathtt{C}} + p_{\mathtt{G}} + p_{\mathtt{T}} = 1 - p_{\mathtt{A}}. \end{aligned} \tag{2.4}$$

Different "runs" of our machine produce strings having different values of $N$. We ultimately wish to know what a "typical" value of $N$ might be, which means we need to know its probability distribution. To find the probability distribution of $N$ is more complicated because we need to know how the individual outputs from our machine are related to each other. This is the topic of the next section.

### 2.3.2 Independence

According to our simple DNA sequence model, the probability distribution of the base in position $r+1$ does not depend on the bases occupying positions $r, \ldots, 2, 1$. This captures the notion that outputs from the machine do not influence each other (you might like to ponder whether this is likely to be true in a DNA sequence). In this section, we formalize the notion of independence for a collection of discrete random variables $X_1, X_2, \ldots, X_n$. Capturing this notion in a definition is a little complicated.

Discrete random variables $X_1, X_2, \ldots, X_n$ are said to be **independent** if, for $r = 2, 3, \ldots, n$,

$$\begin{aligned}\mathbb{P}(X_{i_1} = a_1, X_{i_2} = a_2, \ldots, X_{i_r} = a_r) = \\ \mathbb{P}(X_{i_1} = a_1)\mathbb{P}(X_{i_2} = a_2) \cdots \mathbb{P}(X_{i_r} = a_r)\end{aligned}$$

for all subsets $\{i_1, i_2, \ldots, i_r\}$ of $\{1, 2, \ldots, n\}$ and for all possible values $a_1, \ldots, a_r$. In particular, if $X_1, \ldots, X_n$ are independent, we can calculate the probability of a set of outcomes by using the *multiplication rule for probabilities of independent events*: for events $A_i, i = 1, 2, \ldots, n$, we have

$$\mathbb{P}(X_1 \in A_1, \ldots, X_n \in A_n) = \mathbb{P}(X_1 \in A_1)\mathbb{P}(X_2 \in A_2) \cdots \mathbb{P}(X_n \in A_n). \quad (2.5)$$

For the DNA sequence model outlined in the introduction, the $L_i$ are indeed independent, and so the probability of obtaining the sequence $l_1, l_2, \ldots, l_n$ is given by the product of the probabilities of each $l_i$,

$$\mathbb{P}(L_1 = l_1, \ldots, L_n = l_n) = \mathbb{P}(L_1 = l_1)\mathbb{P}(L_2 = l_2) \cdots \mathbb{P}(L_n = l_n), \quad (2.6)$$

where the terms on the right-hand side are determined by the probability distribution of a single base given in (2.1).

In the last section, we introduced a sequence of discrete random variables $X_1, \ldots, X_n$ that counted whether the bases in the sequence $L_1, L_2, \cdots, L_n$ were `A` or not. It is intuitively clear that if the $L_i$ are independent of one another, then so too are the $X_i$ defined in (2.2). You should check this from the definition. While it is sometimes possible to check whether a collection of random variables $X_1, X_2, \ldots, X_n$ are independent, it is more common to *assume* independence and then use the multiplication rule (2.5) to calculate probabilities of different outcomes. Random variables that are independent and have the same probability distribution are said to be *independent and identically distributed*; in what follows, we use the abbreviation "iid." For further discussion of *dependent* random variables, see Exercise 8.

### 2.3.3 Expected Values and Variances

In this section we describe measures of location (or central tendency) and spread for random variables that take on numerical values. Suppose that $X$ is a discrete random variable taking values in $\mathcal{X}$, a subset of $(-\infty, \infty)$. We define the **expected value** (or *mean* or *expectation*) of $X$ by

$$\mathbb{E}X = \sum_{j=1}^{J} x_j \mathbb{P}(X = x_j) = x_1 p_1 + x_2 p_2 + \cdots + x_J p_J. \tag{2.7}$$

In this book, the symbol $\mathbb{E}$ will always be used to indicate expected value or mean. For the random variables $X_i$ defined in (2.2) with distribution given in (2.4), we have

$$\mathbb{E}X_i = 1 \times p_{\mathtt{A}} + 0 \times (1 - p_{\mathtt{A}}) = p_{\mathtt{A}}. \tag{2.8}$$

If we know the expected value of the random variable $X$, then it is easy to calculate the expected random variable $Y = cX$ for any constant $c$; we obtain

$$\mathbb{E}Y = c\,\mathbb{E}X.$$

The random variable $N$ in (2.3) counts the number of times the base `A` appears in the sequence $L_1, L_2, \ldots, L_n$. We do not yet know the probability distribution of $N$, but we can compute its expected value in another way. We use the fact that the mean of the sum of the $X$s is the sum of the means of the $X$s. That is, for any random variables $X_1, X_2, \ldots, X_n$, we have

$$\mathbb{E}(X_1 + X_2 + \cdots + X_n) = \mathbb{E}X_1 + \mathbb{E}X_2 + \cdots + \mathbb{E}X_n. \tag{2.9}$$

It follows from this and the result in (2.8) that the expected number of times we see an `A` in our $n$ bp sequence is

$$\mathbb{E}N = \mathbb{E}X_1 + \mathbb{E}X_2 + \cdots + \mathbb{E}X_n = n\mathbb{E}X_1 = np_{\mathtt{A}}. \tag{2.10}$$

The expected value of a random variable $X$ gives a measure of its location; values of $X$ tend to be scattered around this value. In addition to this measure of location, we need a measure of spread: Is $X$ closely concentrated about its expected value, or is it spread out? To measure spread, we use a quantity called the **variance**. We define the variance of $X$ by

$$\mathrm{Var}X = \mathbb{E}(X - \mu)^2 = \sum_{j=1}^{J} (x_j - \mu)^2 p_j, \tag{2.11}$$

where $\mu = \mathbb{E}X$ is defined in (2.7). It can be shown that

$$\mathrm{Var}X = \mathbb{E}X^2 - \mu^2 = \sum_{j=1}^{J} x_j^2 p_j - \mu^2, \tag{2.12}$$

a formula that sometimes simplifies calculations. For the random variables $X_i$ in (2.4), we see that

$$\mathrm{Var}X_i = [1^2 \times p_{\mathtt{A}} + 0^2 \times (1 - p_{\mathtt{A}})] - {p_{\mathtt{A}}}^2 = p_{\mathtt{A}}(1 - p_{\mathtt{A}}). \tag{2.13}$$

The (positive) square root of the variance of $X$ is called its **standard deviation** and is denoted by $\mathrm{sd}(X)$. If $X$ is multiplied by the constant $c$, then the variance is multiplied by $c^2$; that is, if $Y = cX$

$$\mathrm{Var}Y = \mathrm{Var}(cX) = c^2\,\mathrm{Var}(X).$$

The standard deviation of $Y$ is then given by

$$\mathrm{sd}Y = \mathrm{sd}(cX) = |c|\,\mathrm{sd}(X).$$

To calculate the variance of the number $N$ of `A` s in our DNA sequence, we exploit the fact that *the variance of a sum of independent random variables is the sum of the individual variances*; that is, if $X_1, \ldots, X_n$ are independent random variables,

$$\mathrm{Var}(X_1 + \cdots + X_n) = \mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_n). \tag{2.14}$$

When this rule is applied to $N$, we find from (2.13) that

$$\mathrm{Var}N = n\mathrm{Var}X_1 = np_{\mathtt{A}}(1 - p_{\mathtt{A}}). \tag{2.15}$$

Equations (2.14) and (2.15) give two statistics that describe features of the probability distribution of $N$ mentioned at the end of Section 2.3.1.

### 2.3.4 The Binomial Distribution

The expected value and variance of a random variable such as $N$ are just two (of many) summary statistics that describe features of its probability distribution. Much more information is provided by the probability distribution itself, which we show how to calculate in this section.

To do this, notice that $\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n)$ is the same for any $x_1, x_2, \ldots, x_n$ containing the same number of 1s. Furthermore, the fact that the $X_i$ are iid means that we can use (2.5) to see that if there are $j$ 1s in $x_1, x_2, \ldots, x_n$, then the probability of that sequence is $p^j(1-p)^{n-j}$, where, for typographical convenience, we have written $p = p_{\mathtt{A}}$. Finally, to compute the probability that the sequence contains $j$ `A` s (i.e., that $N = j$), we need to know how many different realizations of the sequence $x_1, x_2, \ldots, x_n$ have $j$ 1s (and $n - j$ 0s). This is given by the binomial coefficient $\binom{n}{j}$, defined by

$$\binom{n}{j} = \frac{n!}{j!(n-j)!}, \tag{2.16}$$

where $j! = j(j-1)(j-2)\cdots 3\cdot 2\cdot 1$, and by convention $0! = 1$. It now follows that the probability of observing $j$ `A` s is

$$\mathbb{P}(N = j) = \binom{n}{j} p^j (1-p)^{n-j}, j = 0, 1, 2, \ldots, n. \tag{2.17}$$

The probability distribution in (2.17) is known as the **binomial distribution** with parameters $n$ (the number of trials) and $p$ (the probability of success). The mean of $N$, which can also be calculated using (2.17) and (2.7), is given in (2.10), and the variance of $N$ is given in (2.15).

## 2.4 Simulating from Probability Distributions

To understand the behavior of random variables such as $N$, it is useful to simulate a number of instances having the same probability distribution as $N$. If we could get our computer to output numbers $N_1, N_2, \ldots, N_n$ having the same distribution as $N$, we could use them to study the properties of this distribution. For example, we can use the sample mean

$$\overline{N} = (N_1 + N_2 + \cdots + N_n)/n \tag{2.18}$$

to estimate the expected value $\mu$ of $N$. We could use the sample variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(N_i - \overline{N})^2 \tag{2.19}$$

to estimate the variance $\sigma^2$ of $N$, and we can use a histogram of the values of $N_1, \ldots, N_n$ to estimate the probability of different outcomes for $N$.

To produce such a string of observations, we need to tell the computer how to proceed. We need the computer to be able to generate a series of random numbers that we can use to produce $N_1, \ldots, N_n$. This is achieved by use of what are called *pseudo-random numbers*. Many programming languages (and the statistical environment R we use in this book is no exception) have available algorithms (or random number generators) that generate sequences of random numbers $U_1, U_2, \ldots$ that behave as though they are independent and identically distributed, have values in the unit interval $(0, 1)$, and satisfy, for any interval $(a, b)$ contained in $(0, 1)$,

$$\mathbb{P}(a < U_1 \leq b) = b - a.$$

Random variables with this property are said to be *uniformly distributed on* $(0, 1)$. This last phrase captures formally what one understands intuitively: any number between 0 and 1 is a possible outcome, and each is equally likely. Uniform random numbers form the basis of our simulation methods. From now on, we assume we have access to as many such $U$s as we need.

To illustrate the use of our random number generator, we will simulate an observation with the distribution of $X_1$ in (2.2). We can do this by taking a uniform random number $U$ and setting $X_1 = 1$ if $U \leq p \equiv p_{\mathtt{A}}$ and 0 otherwise. This works because the chance that $U \leq p$ is just $p$. Repeating this procedure $n$ times (with a new $U$ each time) results in a sequence $X_1, X_2, \ldots, X_n$ from which $N$ can be computed by adding up the $X$s.

We can use a similar approach to simulate the sequence of bases $L_1, L_2, \ldots, L_n$. This time we divide up the interval (0,1) into four intervals with endpoints at $p_{\mathtt{A}}, p_{\mathtt{A}} + p_{\mathtt{C}}, p_{\mathtt{A}} + p_{\mathtt{C}} + p_{\mathtt{G}}$, and $p_{\mathtt{A}} + p_{\mathtt{C}} + p_{\mathtt{G}} + p_{\mathtt{T}} = 1$. If the simulated $U$ lies in the leftmost interval, set $L_1 = \mathtt{A}$; if it is in the second interval, set $L_1 = \mathtt{C}$; if it is in the third interval, set $L_1 = \mathtt{G}$; and otherwise set $L_1 = \mathtt{T}$. Repeating this procedure with a new value of $U$ each time will produce a sequence of bases $L_1, L_2, \ldots, L_n$. Fortunately, we do not have to write code to implement this approach, as it is included in R already as the `sample` function. The details are given in the box below.

**Computational Example 2.1: Simulating a DNA sequence**

The `sample` function can be used to generate many sorts of samples. The function has the form

```
sample(x,n,replace=TRUE,pi)
   # x = list of values to be sampled
   # n = number of samples required
   # replace=TRUE means sampling with replacement
   # pi = probability distribution for the list in x
```

Here is an application that generates ten outcomes from the probability distribution in (2.4) with $p_A = 0.25$:

```
> pi<-c(0.25,0.75)
> x<-c(1,0)
> sample(x,10,replace=TRUE,pi)
 [1] 1 0 0 0 0 0 1 1 1 0
```

We can use a similar approach to generate a DNA sequence according to our simple iid model. First, we code the bases as $\mathtt{A} = 1$, $\mathtt{C} = 2$, $\mathtt{G} = 3$, and $\mathtt{T} = 4$ and assume that each base is equally likely. To simulate 10,000 bases under this model and look at the first 15 bases, we can use

```
> pi<-c(0.25,0.25,0.25,0.25)
> x<-c(1,2,3,4)
> seq<-sample(x,10000,replace=TRUE,pi)
> seq[1:15]
 [1] 4 4 4 4 1 1 2 3 2 4 2 2 1 1 1
```

It is sometimes convenient to be able to use the same string of random numbers more than once (for example, when preparing examples for this book!). To do this, you can use

```
set.seed(int)
```

where `int` is an integer seed. Try generating two DNA sequences without using `set.seed()` and then by calling `set.seed(100)` before each run. Compare the outputs!

By looking through a given simulated sequence, we can count the number of times a particular pattern arises (for example, the one-letter pattern `A` considered earlier) and so, by repeatedly generating sequences and analyzing each of them, we can get a feel for whether or not our particular pattern is "unusual." We illustrate this by simulating observations having the binomial distribution with $p = 0.25$ and $n = 1000$. Recall that under our uniform base frequency model for DNA, this is the distribution of the number of `A` s in the sequence of length $n$. R can perform binomial simulations, as described in the box below.

**Computational Example 2.2: Simulating binomial random variables**

R has a number of built-in functions for simulating observations from standard distributions. To generate 2000 observations from a binomial distribution with $n = 1000$ trials and success probability $p = 0.25$, we can use

```
> x <- rbinom(2000,1000,0.25)
```

The sample mean (see (2.18)) of our simulated values can be found using

```
> mean(x)
[1] 249.704
```

This value is in good agreement with the mean of $N$, which is $\mu = np = 250$. The variance of the replicates (see (2.19)) can be found using the square of the sample standard deviation:

```
> sd(x)^2
[1] 183.9734
```

Once more, this is in good agreement with the theoretical value of $\sigma^2 = np(1-p) = 187.5$. To plot the histogram, we use

```
> hist(x,xlab="Number of successes",main="")
```

The result is shown in Fig. 2.1.

Later in the book, a number of statistical concepts are introduced by use of **simulation**. For now, we answer another question about the frequency of the pattern `A` in a sequence. Suppose then that we have a sequence of length 1000 bp and assume that each base is equally likely (and so has probability 0.25). How likely is it to observe at least 280 `A` s in such a sequence? There

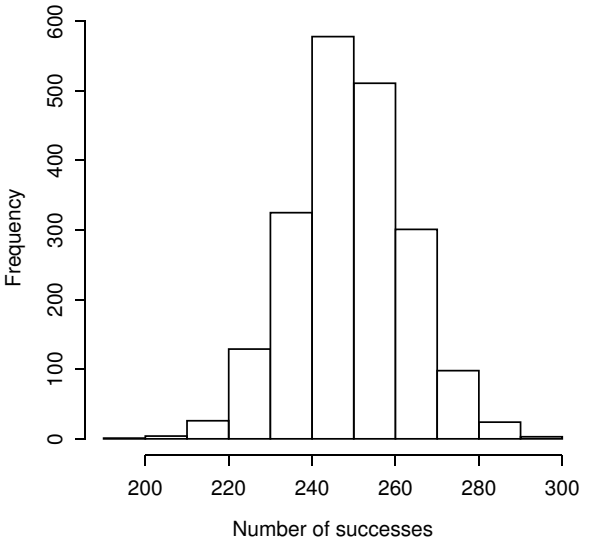


**Fig. 2.1.** Histogram of 2000 replicates of a binomial random variable having $n = 1000$ trials and success probability $p = 0.25$.

are three ways to attack this problem: by using the distribution in (2.17), by simulation, and by an approximation known as the Central Limit Theorem. We discuss the first two approaches here and the third in Section 3.4.

For the first approach, the probability we need to calculate is given by

$$\mathbb{P}(N \geq 280) = \sum_{j=280}^{1000} \binom{1000}{j} (1/4)^j (1 - 1/4)^{1000-j}. \tag{2.20}$$

A computer algebra program such as DERIVE$^{\text{TM}}$ gives the answer 0.01644. The second approach is to simulate a large number of random variables having the distribution of $N$ and calculate how many times the values are greater than or equal to 280. In 10,000 runs of this procedure (see Exercise 4), 149 values were at least 280, so the estimate of the required probability is $149/10,000 \approx 0.015$, in good agreement with the theoretical value of $\approx 0.016$.

## 2.5 Biological Words: $k = 2$

If $l_i$ is a nucleotide at position $i$, then a dinucleotide is $l_i l_{i+1}$ ($5'$ to $3'$ polarity implied). Since $l_i$ is drawn from an alphabet of four bases `A`, `C`, `G`, `T`, there are 16 different dinucleotides: `AA`, `AC`, `AG`, ...,`TG`, `GG`. Since the sum of

the dinucleotide frequencies is 1, just 15 of them suffice to give a complete description of the dinucleotide frequencies in a single-stranded molecule. Dinucleotides are important in part because physical parameters associated with them can describe the trajectory of the DNA helix through space (DNA bending), which may have effects on gene expression. Here we concentrate only on their abundances.

Now suppose we model the sequence $L_1, L_2, \ldots, L_n$ using our iid model with base probabilities given by (2.1). Since the bases are behaving independently of one another, we can use the multiplication rule (2.5) for probabilities to calculate the probabilities of each dinucleotide $r_1 r_2$ as

$$\mathbb{P}(L_i = r_1, L_{i+1} = r_2) = p_{r_1} p_{r_2}. \tag{2.21}$$

For example, under the independence model, the chance of seeing the dinucleotide AA is ${p_{\mathtt{A}}}^2$, and the chance of seeing CG is $p_{\mathtt{C}} p_{\mathtt{G}}$.

To see whether a given sequence has unusual dinucleotide frequencies compared with the iid model, we compare the observed number $O$ of the dinucleotide $r_1 r_2$ with the expected number given by $E = (n-1) p_{r_1} p_{r_2}$. (Note that $n-1$ is the number of dinucleotides in a string of length $n$.) One statistic to use is

$$X^2 = \frac{(O-E)^2}{E}. \tag{2.22}$$

The rationale for using this statistic is as follows. If the observed number is close to the expected number (so that the model is doing a good job of predicting the dinucleotide frequencies), $X^2$ will be small. If, on the other hand, the model is doing a poor job of predicting the dinucleotide frequencies, then $X^2$ will tend to be large.

All that remains is to determine which values of $X^2$ are unlikely if in fact the model is true. This turns out to be a subtle problem that is beyond the scope of this book, but we can at least give a recipe. For further details, see Exercise 10.

(a) Calculate the number $c$ given by

$$c = \begin{cases} 1 + 2p_{r_1} - 3p_{r_1}^2, & \text{if } r_1 = r_2; \\ 1 - 3p_{r_1} p_{r_2}, & \text{if } r_1 \neq r_2. \end{cases}$$

(b) Calculate the ratio $X^2/c$, where $X^2$ is given in (2.22).
(c) If this ratio is larger than 3.84, conclude that the iid model is not a good fit.

If the base frequencies are unknown, the same approach works if the frequencies fr(A), fr(C), fr(G), and fr(T) are estimated from the data. Table 2.2 presents the observed values of $X^2/c$ for the first 1000 bp of two organisms, *E. coli* (GenBank ID NC_000913) and *Mycoplasma genitalium* (GenBank ID L43967). It can be seen that *E. coli* dinucleotide frequencies are not well-described by the simple iid model, whereas the *M. genitalium* sequence is not

as bad. As one might have expected, given the biological nature of genomic sequences, there are some dinucleotides whose frequencies differ significantly from what would be expected from the iid model.

**Table 2.2.** Observed values of $X^2/c$ for the first 1000 bp of each genome. For *E. coli*, the base frequencies were taken as $(0.25, 0.25, 0.25, 0.25)$, whereas for *M. genitalium* they were (0.45, 0.09, 0.09, 0.37), close to the observed frequencies. Significant values are in bold.

| | Observed $X^2/c$ for | |
|---|---|---|
| Dinucleotide | *E. coli* | *M. genitalium* |
| AA | **6.78** | 0.15 |
| AC | 0.05 | 1.20 |
| AG | **5.99** | 0.18 |
| AT | 0.01 | 0.01 |
| CA | 2.64 | 0.01 |
| CC | 0.03 | 0.39 |
| CG | 0.85 | **4.70** |
| CT | **4.70** | 1.10 |
| GA | 2.15 | 0.34 |
| GC | **10.04** | 1.07 |
| GG | 0.01 | 0.09 |
| GT | 1.76 | 0.61 |
| TA | **5.99** | 1.93 |
| TC | **9.06** | 2.28 |
| TG | 3.63 | 0.05 |
| TT | 1.12 | 0.13 |

## 2.6 Introduction to Markov Chains

As we can see from Table 2.2, real genomes have sequence properties more complicated than those described by our simple iid model. A more complicated probabilistic model is required to capture the dinucleotide properties of real genomes. One approach is to use a **Markov chain**, a natural generalization of a sequence of independent trials. Many applications of Markov chains in computational biology are described in Durbin et al. (1998). Suppose that we examine a sequence of letters corresponding to the genome of an organism. If we focus on position $n$, we realize that the character at that position might be dependent upon the letters preceding it. For example, human DNA has a lower than expected frequency of the dinucleotide $5'$-CG-$3'$: if we have a C at position $t-1$, then the probability of a G at position $t$ will be lower than might be expected if the letter at position $t-1$ were A, G, or T. To make these ideas

precise, we make use of more ideas from probability, particularly the notion of *conditional probability*.

### 2.6.1 Conditional Probability

We consider events, which are subsets of the sample space $\Omega$. In the earlier examples, events were usually defined in terms of outcomes of random variables, so that $\Omega$ corresponds to the set $\mathcal{X}$ of possible outcomes of a single experiment. In particular, $\mathbb{P}(\Omega) = 1$, and

$$\mathbb{P}(A) + \mathbb{P}(A^c) = 1,$$

for any event $A$, where $A^c$ denotes the complement $\Omega - A$ of $A$. For two events $A$ and $B$, we define the intersection of $A$ and $B$, written $A \cap B$, as the set of elements in $\Omega$ belonging to both $A$ and $B$. The union of $A$ and $B$, written $A \cup B$, is the set of elements of $\Omega$ belonging to either $A$ or $B$ (and possibly both). The conditional probability of $A$ given $B$, denoted by $\mathbb{P}(A \mid B)$, is defined by

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}, \tag{2.23}$$

when $\mathbb{P}(B) > 0$ (and, by convention, $= 0$ if $\mathbb{P}(B) = 0$). The term $\mathbb{P}(A \cap B)$ is read "probability of $A$ and $B$."

A number of useful consequences derive from this definition, among them Bayes' Theorem, which states that

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \mid B)\mathbb{P}(B)}{\mathbb{P}(A)}. \tag{2.24}$$

Suppose next that $B_1, B_2, \ldots, B_k$ form a *partition* of $\Omega$:

(a) The $B_i$ are disjoint (i.e., $B_i \cap B_j = \emptyset$ for $i \neq j$)
(b) and exhaustive (i.e., $B_1 \cup B_2 \cup \cdots \cup B_k = \Omega$).

Another useful identity is known as the law of total probability: for any event $A$, and a partition $B_1, \ldots, B_k$,

$$\begin{aligned} \mathbb{P}(A) &= \sum_{i=1}^{k} \mathbb{P}(A \cap B_i) \\ &= \sum_{i=1}^{k} \mathbb{P}(A \mid B_i)\mathbb{P}(B_i). \end{aligned} \tag{2.25}$$

A number of applications of these results are given in the exercises at the end of the chapter.

### 2.6.2 The Markov Property

We study a sequence of random variables $\{X_t, t = 0, 1, 2, \ldots\}$ taking values in the state space $\mathcal{X}$. For example, $X_t$ might be the letter in position $t$ of a DNA sequence, and the state space is the set $\mathcal{X} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. The sequence $\{X_t, t \geq 0\}$ is called a *first-order Markov chain* if the probability of finding a particular character at position $t + 1$ given the preceding characters at positions $t, t-1$, and so forth down to position 0 is identical to the probability of observing the character at position $t + 1$ given the character state of the immediately preceding position, $t$. In other words, only the previous neighbor influences the probability distribution of the character at any position. More formally, $\{X_t, t \geq 0\}$ is called a first-order Markov chain if it satisfies the *Markov property*,

$$\mathbb{P}(X_{t+1} = j \mid X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = \mathbb{P}(X_{t+1} = j \mid X_t = i),$$

for $t \geq 0$ and for all $i, j, i_{t-1}, \ldots, i_0 \in \mathcal{X}$. Markov chains of order $k$ correspond to the case where the conditional distribution of the present position depends on the previous $k$ positions. We do not consider higher-order Markov chains in this book.

We consider Markov chains that are homogeneous, which means the probability $\mathbb{P}(X_{t+1} = j \mid X_t = i)$ is independent of the position $t$ in the sequence. For example, $\mathbb{P}(X_{t+1} = \mathtt{G} \mid X_t = \mathtt{C})$ is the same throughout the sequence if the Markov chain is homogeneous. The probabilities common to all positions are denoted by $p_{ij}$,

$$p_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i), \quad i, j \in \mathcal{X}.$$

The $p_{ij}$ are the elements of a matrix $P$ called the *one-step transition matrix* of the chain. In the matrix $P$ below, we show what the transition matrix would look like for DNA. Each row corresponds to one of the possible states at position $t$ (i.e., row 1 corresponds to $X_t = \mathtt{A}$), and each column corresponds to the possible state at $t + 1$ ($X_{t+1} = \mathtt{A}, \mathtt{C}, \mathtt{G}$, or $\mathtt{T}$):

$$P = \begin{pmatrix} p_{\mathtt{AA}} & p_{\mathtt{AC}} & p_{\mathtt{AG}} & p_{\mathtt{AT}} \\ p_{\mathtt{CA}} & p_{\mathtt{CC}} & p_{\mathtt{CG}} & p_{\mathtt{CT}} \\ p_{\mathtt{GA}} & p_{\mathtt{GC}} & p_{\mathtt{GG}} & p_{\mathtt{GT}} \\ p_{\mathtt{TA}} & p_{\mathtt{TC}} & p_{\mathtt{TG}} & p_{\mathtt{TT}} \end{pmatrix}.$$

As we indicated, if the Markov chain is homogeneous, then this transition matrix applies all along the chain. Since after any position there must be one of the characters from $\mathcal{X}$, we see that $\sum_j p_{ij} = 1$ for each value of $i$. If all the rows in $P$ were identical, then the next position from any starting position would have the same distribution, regardless of the identity of the character at the current position. This corresponds to the iid model we used in Section 2.3.2.

The transition matrix tells us the probabilities that apply as we go from the state at position $t$ to the state at position $t+1$. For example, if $X_t = \mathtt{A}$, then the probability that $X_{t+1} = \mathtt{G}$ is $p_{\mathtt{AG}}$. But how do we start the chain? To completely specify the evolution of the Markov chain, we need both the transition matrix and the *initial probability distribution*, $\pi$. The initial probability distribution can be written as a row vector whose elements are

$$\pi_i = \mathbb{P}(X_0 = i), i \in \mathcal{X}.$$

In the case of DNA, the $\pi_i$ are the initial probabilities (at position 0) of `A`, `C`, `G`, and `T`.

To find the probability distribution for the states at position 1 (represented by row vector $\pi^{(1)}$), we use the law of total probability as follows:

$$\begin{aligned}\mathbb{P}(X_1 = j) &= \sum_{i\in\mathcal{X}} \mathbb{P}(X_0 = i, X_1 = j) \\ &= \sum_{i\in\mathcal{X}} \mathbb{P}(X_0 = i)\mathbb{P}(X_1 = j|X_0 = i) \\ &= \sum_{i\in\mathcal{X}} \pi_i p_{ij}. \end{aligned} \tag{2.26}$$

You may recognize this as the product of the row vector $\pi$ with the matrix $P$, so that (in matrix notation)

$$\pi^{(1)} = \pi P.$$

To compute the probability distribution for the states at position 2 (represented by row vector $\pi^{(2)}$), we first note that $\mathbb{P}(X_2 = j|X_0 = i)$ is the $i, j$th element of the matrix $PP = P^2$. This is another application of the law of total probability,

$$\begin{aligned}\mathbb{P}(X_2 = j|X_0 = i) &= \sum_{k\in\mathcal{X}} \mathbb{P}(X_2 = j, X_1 = k|X_0 = i) \\ &= \sum_{k\in\mathcal{X}} \mathbb{P}(X_2 = j|X_1 = k, X_0 = i)\mathbb{P}(X_1 = k|X_0 = i) \\ &= \sum_{k\in\mathcal{X}} \mathbb{P}(X_2 = j|X_1 = k)\mathbb{P}(X_1 = k|X_0 = i) \\ &= \sum_{k\in\mathcal{X}} p_{ik}p_{kj} \\ &= (PP)_{ij}, \end{aligned}$$

as required. (Note that the second line follows from the definition of conditional probability, the third from the Markov property.) Copying the argument that leads to (2.26) then shows that the elements of $\pi^{(2)}$ are given by

$$\pi_j^{(2)} = \mathbb{P}(X_2 = j) = \sum_i \pi_i (P^2)_{ij}.$$

This can be generalized to the $t$th position, giving

$$\mathbb{P}(X_t = j) = \sum_i \pi_i (P^t)_{ij},$$

where the elements $(P^t)_{ij}$ correspond to the elements of the matrix generated by multiplying the transition matrix by itself $t$ times (a total of $t$ factors). This expression gives the probability distribution, $\mathbb{P}(X_t = j)$, at any position $t$.

It is possible that the distribution $\pi^{(t)}$ is the same for every $t$. This is called a **stationary distribution** of the chain. This occurs when

$$\pi_j = \sum_i \pi_i p_{ij} \text{ for all } j.$$

In matrix notation, this condition is $\pi = \pi P$. If $X_0$ also has $\pi$ as its distribution, then $\pi = \pi P^t$ and

$$\mathbb{P}(X_t = j) = \pi_j.$$

This shows that $X_t$ then has the same distribution for every $t$. Note that this does not contradict the dependence of the state at $t$ on the state at $t-1$.

### 2.6.3 A Markov Chain Simulation

To illustrate the use of a Markov chain, we use the observed dinucleotide frequencies of *M. genitalium* to determine the parameters of a Markov model. The observed dinucleotide relative frequencies are given below; each row specifies a base, and each column specifies the following base:

$$\begin{array}{c} \\ \texttt{A} \\ \texttt{C} \\ \texttt{G} \\ \texttt{T} \end{array} \begin{array}{c} \begin{array}{cccc} \texttt{A} & \texttt{C} & \texttt{G} & \texttt{T} \end{array} \\ \begin{pmatrix} 0.146 & 0.052 & 0.058 & 0.089 \\ 0.063 & 0.029 & 0.010 & 0.056 \\ 0.050 & 0.030 & 0.028 & 0.051 \\ 0.087 & 0.047 & 0.063 & 0.140 \end{pmatrix} \end{array}. \tag{2.27}$$

The individual base frequencies (the base composition) may be calculated from the matrix by summing across the rows. We obtain $p_{\texttt{A}} = 0.345, p_{\texttt{C}} = 0.158, p_{\texttt{G}} = 0.159$, and $p_{\texttt{T}} = 0.337$. To estimate the AA element of the transition matrix $P$ of the chain, we note that

$$p_{\texttt{AA}} = \mathbb{P}(X_t = \texttt{A} \mid X_{t-1} = \texttt{A}) = \frac{\mathbb{P}(X_t = \texttt{A}, X_{t-1} = \texttt{A})}{\mathbb{P}(X_{t-1} = \texttt{A})} \approx \frac{0.146}{0.345} = 0.423.$$

In this calculation, we used the observed dinucleotide frequency matrix to find the proportion 0.146 of the dinucleotide AA and the base frequency vector to find the proportion 0.345 of A. The same estimation procedure can be done

for the 15 other entries of $P$, and we arrive at an estimated transition matrix of

$$P = \begin{array}{c} \\ \texttt{A} \\ \texttt{C} \\ \texttt{G} \\ \texttt{T} \end{array} \begin{array}{c} \begin{array}{cccc} \texttt{A} & \texttt{C} & \texttt{G} & \texttt{T} \end{array} \\ \begin{pmatrix} 0.423 & 0.151 & 0.168 & 0.258 \\ 0.399 & 0.184 & 0.063 & 0.354 \\ 0.314 & 0.189 & 0.176 & 0.321 \\ 0.258 & 0.138 & 0.187 & 0.415 \end{pmatrix} \end{array}.$$

The rows sum to unity, to within rounding error, as they should. The smallest matrix element ($p_{\texttt{CG}} = 0.063$) corresponds to placing `G` at a position given `C` at the previous position. For our initial distribution $\pi$, we assign vector elements just using the base composition:

$$\pi = (0.345, 0.158, 0.159, 0.337).$$

Now we are ready to simulate a sequence string that resembles *M. genitalium* DNA, at least in terms of dinucleotide frequencies. We don't expect it to look like actual *M. genitalium* DNA because our probabilistic model is still likely to be too simple: the model is capable of generating the correct proportions of $k$-tuples with $k = 1$ and $k = 2$, but it does not include the "machinery" for simulating $k$-tuple frequencies for $k > 2$. The details are given in Computational Example 2.3.

**Computational Example 2.3: Simulating a string having characteristics of *Mycoplasma* DNA**

We generate a sequence having 50,000 positions. We code the bases as follows: $\texttt{A} = 1, \texttt{C} = 2, \texttt{G} = 3$, and $\texttt{T} = 4$. By using numbers instead of characters, we can use logical operators (`==, !=, >, ...`) in our analysis of the sequence. The values for the transition matrix and $\pi$ were presented above. We simulate the sequence with the aid of R (Appendix A). First, we write an R function (program) to generate the sequence. For this function, we supply the following input variables (arguments): the transition matrix $P$, $\pi$, and $n$, the length of the string to be generated. We also supply a vector $x$ containing the characters to be sampled. A function that will simulate the sequence is presented below:

```
> markov1 <- function(x,pi,P,n){
 # x = vector [1 2 3 4] representing A, C, G, T, respectively
 # pi = the probability distribution for X0: (1x4 row vector)
 # P = transition matrix (4x4)
 # n = length of simulated sequence
 # Initialize vector to contain simulated sequence
 mg <- rep(0,n)
 # Produce initial element
 mg[1] <- sample(x,1,replace=TRUE,pi)
 for(k in 1:(n-1)){
```

```
mg[k+1]<-sample(x,1,replace=T,P[mg[k],])
                    }
return(mg)
}
```

Lines prefixed by # are comments and are not executed. The R library function `sample()` is employed to generate an element to be placed at position $i+1$ given a particular letter at $i$. Use the `help(sample)` command at the R prompt for documentation for this function. Note particularly how the probability distributions `pi` and `P[i,]`, the rows of the transition matrix, are employed for each use of `sample()`. Each application of `markov1` will produce a different string (check this), but the overall properties of each string should be similar. To input the parameters in the simulation, we use:

```
> x <- c(1:4)                  # Loading parameters
> pi <- c(.342,.158,.158,.342)
> P <- matrix(scan(),ncol=4, nrow=4,byrow=T)
1:  .423 .151 .168 .258
5:  .399 .184 .063 .354
9:  .314 .189 .176 .321
13: .258 .138 .187 .415
17:          # enter "return" here to end input
```

Application of `markov1` uses the following:

```
> tmp<-markov1(x,pi,P,50000)
```

**Checking the simulation output**

We can check the base composition (remembering that `C` is represented by 2 and `G` is represented by 3) of the generated sequence:

```
> length(tmp[tmp[]==1])
[1] 16697
> length(tmp[tmp[]==2])
[1] 8000
> length(tmp[tmp[]==3])
[1] 7905
> length(tmp[tmp[]==4])
[1] 17398
> (8000+7905)/(16697+8000+7905+17398) # compute fr(G+C)
[1] 0.3181
```

This compares favorably with the value 31.6% `G+C` given in the transition matrix. Now we check whether `tmp` contains an appropriate fraction of `CG` dinucleotides:

```
> count=0
> for(i in 1:49999){  # 49999 because i+1 undefined for 50000
+ if(tmp[i]==2 && tmp[i+1]==3)
+ count<-count+1}
> count
[1] 482
> count/49999
[1] 0.0096
```

From (2.27), the relative abundance of the CG dinucleotide in *M. genitalium* is 0.010, whereas the string produced by the Markov model contains CG at a relative abundance 0.0096. This matches the observed data well. You should verify that other dinucleotide relative abundances are correctly predicted by your simulation. Evidently, the Markov model provides a good probabilistic description of the data for *M. genitalium* DNA.

## 2.7 Biological Words with $k = 3$: Codons

As mentioned in Chapter 1, there are 61 codons that specify amino acids and three stop codons. Since there are 20 common amino acids, this means that most amino acids are specified by more than one codon. This has led to the use of a number of statistics to summarize the "bias" in codon usage. An example of such a statistic is shown later. To show how these codon frequencies can vary, consider the specific example of the *E. coli* proteins. Table 2.3 displays the predicted and observed codon relative frequencies for three (out of 20) particular amino acids found in 780 genes of *E. coli*. (At the time this work was done, no complete genome sequences were available for any organism.) The predicted relative frequencies are calculated as follows.

For a sequence of independent bases $L_1, L_2, \ldots, L_n$ the expected 3-tuple relative frequencies can be found by using the logic employed for dinucleotides in (2.21). We calculate the probability of a 3-word as

$$\begin{aligned}&\mathbb{P}(L_i = r_1, L_{i+1} = r_2, L_{i+2} = r_3) = \\ &\quad \mathbb{P}(L_i = r_1)\mathbb{P}(L_{i+1} = r_2)\mathbb{P}(L_{i+2} = r_3).\end{aligned} \tag{2.28}$$

This provides the expected frequencies of particular codons. To get the entries in Table 2.3, we calculate the relative proportion of each of the codons making up a particular amino acid. Using the base frequencies from Table 2.1, we find that

$$\mathbb{P}(\texttt{TTT}) = 0.246 \times 0.246 \times 0.246 = 0.01489,$$

while

$$\mathbb{P}(\texttt{TTC}) = 0.246 \times 0.246 \times 0.254 = 0.01537.$$

It follows that among those codons making up the amino acid Phe, the expected proportion of TTT is

$$\frac{0.01489}{0.01489 + 0.01537} = 0.492.$$

Allowing for approximations in the base frequencies of *E. coli*, this is the value given in the first row of the second column in Table 2.3.

**Table 2.3.** Comparison of predicted and observed triplet frequencies in coding sequences for a subset of genes and codons from *E. coli*. Figures in parentheses below each gene class show the number of genes in that class. Data were taken from Médigue et al. (1991).

| | | | Observed | |
|---|---|---|---|---|
| | Codon | Predicted | Gene Class I (502) | Gene Class II (191) |
| Phe | TTT | 0.493 | 0.551 | 0.291 |
| | TTC | 0.507 | 0.449 | 0.709 |
| Ala | GCT | 0.246 | 0.145 | 0.275 |
| | GCC | 0.254 | 0.276 | 0.164 |
| | GCA | 0.246 | 0.196 | 0.240 |
| | GCG | 0.254 | 0.382 | 0.323 |
| Asn | AAT | 0.493 | 0.409 | 0.172 |
| | AAC | 0.507 | 0.591 | 0.828 |

Médigue et al. (1991) clustered the different genes into three groups based on such codon usage patterns, and they observed three clusters. For Phe and Asn different usage patterns are observed for Gene Class I and Gene Class II. For Gene Class II in particular, the observed codon frequencies differ considerably from their predicted frequencies. When Médigue et al. checked the gene annotations (names and functions), they found that Class II genes were largely those such as ribosomal proteins or translation factors—genes expressed at high levels—whereas Class I genes were mostly those that are expressed at moderate levels.

A statistic that can describe each protein-coding gene for any given organism is the **codon adaptation index**, or CAI (Sharp and Li, 1987). This statistic compares the distribution of codons actually used in a particular protein with the preferred codons for highly expressed genes. (One might also compare them to the preferred codons based on gene predictions for the whole genome, but the CAI was devised prior to the availability of whole-genome sequences.) Consider a sequence of amino acids $X = x_1, x_2, \ldots, x_L$ representing protein $X$, with $x_k$ representing the amino acid residue corresponding to codon $k$ in the gene. We are interested in comparing the actual codon usage

with an alternative model: that the codons employed are the most probable codons for highly expressed genes. For the codon corresponding to a particular amino acid at position $k$ in protein $X$, let $p_k$ be the probability that *this* particular codon is used to code for the amino acid in highly expressed genes, and let $q_k$ correspond to the probability for *the most frequently used* codon of the corresponding amino acid in highly expressed genes. The CAI is defined as

$$\text{CAI} = \left[ \prod_{k=1}^{L} p_k/q_k \right]^{1/L}.$$

In other words, the CAI is the geometric mean of the ratios of the probabilities for the codons *actually* used to the probabilities of the codons *most frequently* used in highly expressed genes. An alternative way of writing this is

$$\log(\text{CAI}) = \frac{1}{L} \sum_{k=1}^{L} \log(p_k/q_k).$$

This expression is in terms of a sum of the logarithms of probability ratios, a form that is encountered repeatedly in later contexts.

For an example of how this works, consider the amino acid sequence from the amino terminal end of the *himA* gene of *E. coli* (which codes for one of the two subunits of the protein IHF: length $L = 99$). This is shown in Fig. 2.2, and below it are written the codons that appear in the corresponding gene. Underneath is a table showing probabilities (top half) and corresponding codons (in corresponding order) in the bottom half. The maximum probabilities (the $q_k$) are underlined. The CAI for this fragment of coding sequence is then given by

$$\text{CAI} = \left[ \frac{1.000}{1.000} \times \frac{0.199}{0.469} \times \frac{0.038}{0.888} \times \frac{0.035}{0.468} \cdots \right]^{1/99}.$$

The numerator of each fraction corresponds to $p_k$, the probability that the *observed* codon in the *himA* gene sequence would actually be used in a highly expressed gene. If every codon in a gene corresponded to the most frequently used codon in highly expressed genes, then the CAI would be 1.0. In *E. coli*, a sample of 500 protein-coding genes displayed CAI values in the range from 0.2 to 0.85 (Whittam, 1996).

Why do we care about statistics such as the CAI? As we will see in Chapter 11, there is a correlation between the CAI and mRNA levels. In other words, the CAI for a gene sequence in genomic DNA provides a first approximation of its expression level: if the CAI is relatively large, then we would predict that the expression level is also large.

If we wanted a probabilistic model for a genome, $k = 3$, we could employ a second (or higher)-order Markov chain. In the second-order model, the state at position $t + 1$ depends upon the states at both $t$ and $t - 1$. In this case, the transition matrix could be represented by using 16 rows (corresponding to all

| M | A | L | T | K | A | E | M | S | E | Y | L | F | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ATG | GCG | CTT | ACA | AAA | GCT | GAA | ATG | TCA | GAA | TAT | CTG | TTT | ... |
| 1.000 | 0.469 | 0.018 | 0.451 | 0.798 | 0.469 | 0.794 | 1.000 | 0.428 | 0.794 | 0.193 | 0.018 | 0.228 | |
| | 0.057 | 0.018 | 0.468 | 0.202 | 0.057 | 0.206 | | 0.319 | 0.206 | 0.807 | 0.018 | 0.772 | |
| | 0.275 | 0.038 | 0.035 | | 0.275 | | | 0.033 | | | 0.038 | | |
| | 0.199 | 0.033 | 0.046 | | 0.199 | | | 0.007 | | | 0.033 | | |
| | | 0.007 | | | | | | 0.037 | | | 0.007 | | |
| | | 0.888 | | | | | | 0.176 | | | 0.888 | | |
| ATG | GCT | TTA | ACT | AAA | GCT | GAA | ATG | TCT | GAA | TAT | TTA | TTT | |
| | GCC | TTG | ACC | AAG | GCC | GAG | | TCC | GAG | TAC | TTG | TTC | |
| | GCA | CTT | ACA | | GCA | | | TCA | | | CTT | | |
| | GCG | CTC | ACG | | GCG | | | TCG | | | CTC | | |
| | | CTA | | | | | | AGT | | | CTA | | |
| | | CTG | | | | | | AGC | | | CTG | | |

**Fig. 2.2.** Example of codon usage patterns in *E. coli* for computation of the codon adaptation index of a gene. The probability for the most frequently used codon in highly expressed genes is underlined.

possible dinucleotide states for $t-1$ and $t$) and four columns (corresponding to the possible states at position $t+1$). We do not explore this further here.

## 2.8 Larger Words

The number and distributions of $k$-tuples, $k > 3$, can have practical and biological significance. Some particularly important $k$-tuples correspond to $k = 4, 5, 6$, or 8. These include recognition sites for restriction endonucleases (e.g., 5′-AGCT-3′ is the recognition sequence for endonuclease *Alu*I, 5′-GAATTC-3′ is the recognition sequence for *Eco*RI, and 5′-GCGGCCGC-3′ is the recognition sequence for *Not*I). The distribution of these $k$-tuples throughout the genome will determine the distribution of restriction endonuclease digest fragments ("restriction fragments"). These distributions are discussed in Chapter 3. There are also particular words (e.g., Chi sequences 5′-GCTGGTGG-3′ in *E. coli*, $k = 8$) that may be significantly over-represented in particular genomes or on one or the other strands of the genome. For example, Chi appears 761 times in the *E. coli* chromosome compared with approximately 70 instances predicted using base frequencies under the iid model. Moreover, Chi sequences are more abundant on the leading strand than on the **lagging strand**. These observations relate in part to the involvement of Chi sequences in generalized recombination. Another example is the uptake sequences that function in bacterial transformation (e.g., 5′-GCCGTCTGAA-3′ in *Neisseria gonorrhoeae*, $k = 10$). Other examples of over-represented $k$-tuples can be found in the review by Karlin et al. (1998). Some sequences may be under-represented. For example, 5′-CATG-3′ occurs in the *E. coli* K-12 chromosome at about 1/20 of the expected frequency.

$k$-words ($k \geq 4$) are also useful for analyzing particular genomic subsequences. At the end of the next chapter, we illustrate how 4-word frequencies can be used to quantify the differences between *E. coli* promoter sequences and "average" genomic DNA.

## 2.9 Summary and Applications

In the cases $k = 1$, 2, and 3 above, we saw that frequencies of words or statistics derived from them (GC skew for $k = 1$) were not as predicted from the independent, identically distributed base model. This is no surprise: genomes code for biological information, and we would therefore not expect the iid model to provide an accurate description for real genomes. The frequencies of $k$-tuples have a number of applications. We already mentioned that GC skew can be used to predict locations of replication origins and termini in prokaryotes. Prokaryotes also may engage in gene transfer, and local genome regions having aberrant base compositions may indicate genome segments acquired by lateral transfer. For eukaryotes, gene regions may have on average a different base composition than regions outside genes (e.g., human genes are relatively GC-rich compared with the genome as a whole).

For $k = 3$, we saw that different gene classes have different codon usage frequencies. In general, the distribution of codon usage differs from organism to organism. The codon usage pattern of an anonymous DNA sequence from an organism can be compared against the overall usage pattern for that organism to help determine whether the reading frame being analyzed probably is, or is not, a protein-coding region. In Sections 2.5 and 2.7, words were described in terms of probabilistic models. Sometimes the observed frequencies of $k$-words can be used to make inferences about DNA sequences. For example, suppose that we were given a sequence string that hypothetically could be a portion of a candidate exon or prokaryotic gene:

GACGTTAGCTAGGCTTTAATCCGACTAAACCTTTGATGCATGCCTAGGCTG

Simply by noting the stop codons (underlined) in all three reading frames, and knowing that a typical bacterial gene contains, on average, more than 300 codons, or that the typical human exon, for example, contains around 50 codons, we can make a reasonable inference that this string does not code for a protein.

$k$-tuple frequencies can assist in classifying DNA sequences by content, such as predicting whether an unannotated sequence is coding or noncoding. Because coding sequences commonly specify amino acid strings that are functionally constrained, we would expect that their distribution of $k$-tuple frequencies would differ from that of noncoding sequences (e.g., intergenic or intronic sequences). Consider in-frame hexamers ($k = 6$). There are 4096 of these 6-tuple words. We can already predict from known polypeptide sequence

data that some 6-tuples will be infrequent. For example, the pair of residues Trp-Trp in succession is not common in protein sequences, which implies that the corresponding dicodon hexamer, TTGTTG, is likely to be relatively infrequent in coding sequences. Alternatively, we could use $k = 3$ and compute the CAI within open reading frames to identify those that might correspond to highly expressed genes (i.e., CAI close to 1.0). $k$-tuple frequencies and other content measures such as the presence of particular signals (see Chapter 9) are among the statistical properties employed by computational gene-finding tools.

## References

Campbell AM, Mrázek J, Karlin S (1999) Genome signature comparisons among prokaryote, plasmid, and mitochondrial DNA. *Proceedings of the National Academy of Sciences USA* 96:9184–9189.

DERIVE™ 5. See `http://education.ti.com/us/product/software/derive/features/features.html`

Durbin R, Eddy S, Krogh A, Mitchison G (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge: Cambridge University Press.

Karlin S, Campbell AM, Mrázek J (1998) Comparative DNA analysis across diverse genomes. *Annual Review of Genetics* 32:185–225.

Médigue C, Rouxel T, Vigier P, Hénaut A, Danchin A (1991) Evidence for horizontal gene transfer in *Escherichia coli* speciation. *Journal of Molecular Biology* 222:851–856.

Sharp PM, Li W-H (1987) The codon adaptation index—a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research* 15:1281–1295.

Whittam TS (1996) In Neidhardt FC (Ed. in Chief), Curtiss III R, Ingraham JL, Lin ECC, Low KB, Magasanik B, Reznikoff WS, Riley M, Schaechter M, Umbarger HE (eds). *Escherichia coli and Salmonella: Cellular and Molecular Biology*. Washington, D.C.: ASM Press, pp 2708–2720.

## Exercises

**Exercise 1.** The base composition of a certain microbial genome is $p_{\mathtt{G}} = p_{\mathtt{C}} = 0.3$ and $p_{\mathtt{A}} = p_{\mathtt{T}} = 0.2$. We are interested in 2-words where the letters are assumed to be independent. There are $4 \times 4 = 16$ 2-words.

(a) Present these 16 probabilities in a table. (Do your 16 numbers sum to 1.0?)

(b) Purine bases are defined by $\mathtt{R} = \{\mathtt{A}, \mathtt{G}\}$ and pyrimidine bases by $\mathtt{Y} = \{\mathtt{C}, \mathtt{T}\}$. Let $E$ be the event that the first letter is a pyrimidine, and $F$ the event

that the second letter is `A` or `C` or `T`. Find $\mathbb{P}(E), \mathbb{P}(F), \mathbb{P}(E \cup F), \mathbb{P}(E \cap F)$, and $\mathbb{P}(F^c)$.

(c) Set G=`{CA, CC}`. Calculate $\mathbb{P}(G \mid E)$, $\mathbb{P}(F \mid G \cup E)$, $\mathbb{P}(F \cup G \mid E)$.

**Exercise 2.** For three events $A, B$, and $C$ show that $\mathbb{P}(A \cap B \mid C) = \mathbb{P}(A \mid B \cap C)\mathbb{P}(B \mid C)$.

**Exercise 3.** The independent random variables $X$ and $Y$ have the following expectations: $\mathbb{E}(X) = 3, \mathbb{E}(X^2) = 12, \mathbb{E}(Y) = 5$, and $\mathbb{E}(Y^2) = 30$. Find

(a) $\mathbb{E}(X+Y), \mathbb{E}(2X+1), \mathbb{E}(2X+0.3Y)$ and $\mathbb{E}(2X-0.3Y)$.
(b) $\mathrm{Var}X, \mathrm{Var}Y, \mathrm{Var}(2X+5)$, $\mathrm{Var}(X+Y), \mathrm{Var}(5X+7Y), \mathrm{Var}(5X-7Y)$, and $\mathrm{Var}(5X+7Y+1600)$.

**Exercise 4.** Suppose $N$ has a binomial distribution with $n = 10$ and $p = 0.3$.

(a) Using the formula (2.17), calculate $\mathbb{P}(N=0), \mathbb{P}(N=2), \mathbb{E}(N)$, and $\mathrm{Var}N$.
(b) Using R and Computational Example 2.2, simulate observations from $N$. Use the simulated values to estimate the probabilities you calculated in (a), and compare with the results in (a).
(c) Now use R to simulate observations from $N$ when $n = 1000$ and $p = 0.25$. What is your estimate of $\mathbb{P}(N \geq 280)$? (See (2.20).)

**Exercise 5.** Verify the terms in the first row of the transition matrix $P$ presented in Section 2.6.3. Describe how you would use the sequence of *M. genitalium* to produce this matrix.

**Exercise 6.** Find the stationary distribution of the chain with transition matrix $P$ in Section 2.6.3; that is, solve the equations $\pi = \pi P$ subject to the elements of $\pi$ begin positive and summing to 1. Compare $\pi$ to the base composition of *M. genitalium*, and comment.

**Exercise 7.** Using the values for $P$ in Section 2.6.3, compute $P^2$, $P^4$ and $P^8$. (Remember to use `%*%` as the matrix multiplication operator in R, not `*`.) What quantities are the row entries approaching? This distribution is called the *limiting distribution* of the chain. Compare to the results of Exercise 6.

**Exercise 8.** Perform the simulation in Chapter 2.6.3, and verify that the appropriate dinucleotide frequencies are produced in your simulated string.

**Exercise 9.** Using the sequence of *E. coli* (GenBank ID `NC_000913`) and the method in Section 2.6.3, find the dinucleotide frequencies and the estimated transition matrix. (Hint: Download the sequence in FASTA format from the NCBI website, and convert letters to numbers using a text editor.)

**Exercise 10.** In this example, we use R to verify the distribution of the statistic $X^2$ given in (2.22), as used in Table 2.2. To do this, first choose a pair of bases $r_1 r_2$, and calculate the appropriate value of $c$ by following the recipe after (2.22) for the given base frequencies $p = (p_1, \ldots, p_4)$. Now use R to

repeatedly simulate strings of 1000 letters having distribution $p$, calculate $O$ (the number of times the pair of letters $r_1 r_2$ is observed) and $E$, and hence $X^2/c$. Plot a histogram of these values, and compare it to the theoretical distribution (which is the $\chi^2$ distribution with 1 degree of freedom). Remark: This simulation approach provides a useful way to estimate percentage points of the distribution of *any* test statistic.

**Exercise 11.** The genome composition $\pi$ of *E. coli* can be computed from Table 2.1. Take the first 1000 bps of the *E. coli* sequence you used in the previous exercise. We are going to use a variant of (2.22) to test if this 1000 bp sequence has an unusual base composition when compared with $\pi$. The statistic to use is

$$X^2 = \sum_{i=1}^{4} \frac{(O_i - E_i)^2}{E_i}, \tag{2.29}$$

where $O_i$ denotes the number of times base $i$ is observed in the sequence, and $E_i$ denotes the expected number (assuming that frequencies are given by $\pi$).

(a) Calculate $O_i$ and $E_i, i = 1, \ldots, 4$, and then $X^2$.
(b) Values of $X^2$ that correspond to unusual base frequencies are determined by the large values of the $\chi^2$ distribution with $4-1 = 3$ degrees of freedom. Using a 5% level of significance, are data consistent with $\pi$ or not? [Hint: percentage points of the $\chi^2$ distribution can be found using R.]

**Exercise 12.** In this exercise we have two random variables $X$ and $Y$ which are not independent. Their joint probability distribution is given in the following table:

| | | $Y$ | | | |
|---|---|---|---|---|---|
| | | 1 | 3 | 6 | 9 |
| $X$ | 2 | 0.11 | 0.05 | 0.20 | 0.08 |
| | 3 | 0.20 | 0.02 | 0.00 | 0.10 |
| | 7 | 0.00 | 0.05 | 0.10 | 0.09 |

The values of $X$ are written in the first column and the values of $Y$ in the first row. The table is read as $\mathbb{P}(X = 7 \,\&\, Y = 6) = 0.10$, and so on.

(a) Find the marginal distribution of $X$ and $Y$. (That is, $\mathbb{P}(X = 2), \mathbb{P}(X = 3), \ldots$.)
(b) Write $Z = XY$. Find the probability distribution of $Z$.
(c) The **covariance** between any two random variables is defined by

$$\mathrm{Cov}(X, Y) = \mathbb{E}(X - \mathbb{E}X)(Y - \mathbb{E}Y).$$

Show that $\mathrm{Cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}X \times \mathbb{E}Y$.
(d) Find $\mathbb{E}X, \mathbb{E}Y, \sigma_X^2 = \mathrm{Var}X, \sigma_Y^2 = \mathrm{Var}Y$, and $\mathrm{Cov}(X, Y)$ for the example in the table.

(d) The **correlation coefficient** $\rho$ is defined by $\rho_{X,Y} = \mathrm{Cov}(X,Y)/\sigma_X\sigma_Y$. It can be shown that $-1 \le \rho \le 1$, the values $\pm 1$ arising when $Y$ is a linear function of $X$. Verify this last statement.
(e) Calculate $\rho$ for the example in the table.

**Exercise 13.** Using R, simulate $n$ pairs of observations $(X_i, Y_i), i = 1, 2, \ldots, n$ from the distribution in the table in Exercise 12.

(a) From these observations calculate the estimates $\bar{X}, \bar{Y}, s_X^2$, and $s_Y^2$ (see (2.18), (2.19)).
(b) Calculate the estimate $s_{X,Y}^2$ of $\mathrm{Cov}(X,Y)$ defined by

$$s_{X,Y}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}).$$

(c) Calculate the estimate $r$ of the correlation coefficient $\rho$ via $r = s_{X,Y}^2/(s_X\, s_Y)$.
(d) Compare the estimated covariance and correlation obtained for different values of $n$ with the theoretical values obtained in Exercise 12.