Multiscale Analysis of Complex Time Series: Integration of Chaos and Random Fractal Theory, and Beyond
by Jianbo Gao, Yinhe Cao, Wen-wen Tung and Jing Hu
Copyright © 2007 John Wiley & Sons. Inc.

CHAPTER 8

LONG MEMORY PROCESSES AND STRUCTURE-FUNCTION–BASED MULTIFRACTAL ANALYSIS

In Chapter 6, we introduced fractional Brownian motion (fBm) model and briefly talked about persistent and antipersistent correlations in a time series. Persistent and antipersistent correlations may be collectively called longmemory. In this chapter, we describe the general theory of longmemory in a time series and discuss various methods of quantifying longmemory. We then introduce structure-function–based multifractal analysis, which provides a more comprehensive characterization of a complex time series than the concept of longmemory. To appreciate the power as well as the limitations of the concepts and methodologies, we shall discuss a number of different models as well as applications. Special attention is paid to two important notions, fractal scaling break and consistency of different methods of quantifying memory, as well as multifractal properties of a time series data. We shall also examine the meaning of dimension reduction in a fractal time series using principal component analysis.

8.1 LONG MEMORY: BASIC DEFINITIONS

Let $X = \{X_t : t = 0, 1, 2, \dots\}$ be a covariance stationary stochastic process with mean μ, variance σ^2, and autocorrelation function $r(k), k \geq 0$. Assume $r(k)$ to be of the form

$$r(k) \sim k^{2H-2} \ as \ k \to \infty, \tag{8.1}$$

where $0 < H < 1$ is the Hurst parameter and measures the persistence of the correlation:

- When $0 < H < 1/2$, the process is said to have antipersistent correlation.
- When $1/2 < H < 1$, the process has persistent correlation; the larger the H value, the more persistent the correlation is. In this case, we have

  $$\sum_k r(k) = \infty.$$

  Because of this property, the time series is said to be long-range-dependent (LRD).
- When $H = 1/2$, the time series is said to be either memoryless or short-range-dependent (SRD).

Do processes with persistent and antipersistent correlations exist? The answer is yes. For example, the hydrologist Hurst found that the water level of Niles has an $H \approx 0.74$. For real network traffic, one often finds $0.7 \le H \le 0.9$. The best-known example of antipersistence is perhaps Kolmogorov's energy spectrum of turbulence with $H = 1/3$ (this will be discussed in some depth in Sec. 9.4). In Chapter 6, we discussed a simple model, fractional Gaussian noise (fGn), and showed in Eq. (6.24) that the autocorrelation function, $r(k)$, for an fGn process satisfies the following equation:

$$\lim_{k\to\infty} \frac{r(k)}{k^{2H-2}} = H(2H-1). \tag{8.2}$$

Hence, an fGn process is an example of a process with longmemory, and as noted in the following, it is also an *exact* longmemory process in a strict mathematical sense.

Next, we construct a new covariance stationary time series

$$X^{(m)} = \{X_t^{(m)} : t = 1, 2, 3, \dots\}, \ m = 1, 2, 3, \dots,$$

obtained by averaging the original series X over nonoverlapping blocks of size m,

$$X_t^{(m)} = (X_{tm-m+1} + \cdots + X_{tm})/m, \ \ t \ge 1\,. \tag{8.3}$$

There are several useful relationships between the autocorrelation functions of the original LRD process and its averaged version. Using the stationarity properties of the processes, a general formula for the autocorrelation function, $r^{(m)}(k)$, of $X^{(m)}$, can be stated as

$$r^{(m)}(k) = \frac{(k+1)^2 V_{(k+1)m} - 2k^2 V_{km} + (k-1)^2 V_{(k-1)m}}{2V_m}, \tag{8.4}$$

where $V_m = var(X^{(m)})$. Using this relationship, it is straightforward to verify that the variance of $X^{(m)}$ satisfies

$$var(X^{(m)}) = \sigma^2 m^{2H-2} \tag{8.5}$$

if and only if the autocorrelation function of the LRD process satisfies

$$r(k) = \frac{1}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right]. \tag{8.6}$$

Moreover, one can verify that if X satisfies Eq. (8.5), then the autocorrelation function, $r^{(m)}(k)$, of the process $X^{(m)}$ satisfies

$$r^{(m)}(k) = r(k), \quad k \geq 0\,. \tag{8.7}$$

A process X that satisfies Eq. (8.5) (or equivalently, Eqs. (8.1), (8.5), and (8.7)) is often referred to as an exactly second-order self-similar process. Note that such a process always satisfies Eq. (8.2). On the other hand, instead of satisfying Eq. (8.5), if one has

$$\lim_{k\to\infty} \frac{r(k)}{k^{2H-2}} = c_1,$$

where $0 < c_1$ is an arbitrary constant, then one can show that

$$\lim_{k\to\infty} \frac{var(X^{(m)})}{m^{2H-2}} = c_2 \tag{8.8}$$

for some constant $c_2 > 0$. Such a process is often referred to as an asymptotically second-order self-similar process.

Equation (8.5) (or, more generally, Eq. (8.8)) is often called the variance-time relation. It provides a simple and precise way of quantifying the "little smoothing" behavior depicted in Fig. 6.4. For ease of exposition, let us assume that $H = 0.75$ for real traffic, and at the smallest time scale, real traffic and Poisson traffic have the same variance σ_0^2. For Poisson traffic, since $H = 0.5$, $var(X_i^{(m)})$ drops to $10^{-2}\sigma_0^2$ when $m = 100$. When $H = 0.75$, however, we need $m = 10,000$ to have the same effect. That is exactly what Fig. 6.4 has shown us.

To further illustrate the significance of the Hurst parameter, let us digress to consider a wellknown physical phenomenon: scattering of light. As is well-known, a harmonic oscillator may be described by $A_1 \cos(\omega t + \phi_1)$. Then the intensity of light is $\sim A_1^2$. When there are two oscillators described as $A_1 \cos(\omega_1 t + \phi_1)$ and $A_2 \cos(\omega t + \phi_2)$, the intensity of light is $\sim A_1^2 + A_2^2 + 2A_1A_2\cos(\phi_1 - \phi_2)$. When the phase difference is fixed, we have interference; otherwise, there is no interference. This corresponds to averaging out of the cross term. When there are m oscillators, under the condition $A_i = A = \text{const}$, if there is no interference, then the intensity of light is $\sim mA^2 = m^{2H}A^2$, $H = 1/2$; when interference is maximal, the intensity of light is $\sim (mA)^2 = m^{2H}A^2$, $H = 1$. This is how a laser works.

8.2 ESTIMATION OF THE HURST PARAMETER

Using a proof similar to that developed in Sec. 6.1, one can prove that Eq. (8.1) implies power-law power spectral density (PSD) for the process x, $1/f^{2H-1}$, $f \to 0$. Therefore, processes with longmemory are a type of $1/f$ processes. Note the difference between this expression and Eq. (6.7). The latter is for the PSD of a corresponding random walk process, e.g., for fBm that is a random walk process generated by integrating the increment process fGn.

Due to the ubiquity of longmemory processes, estimation of the Hurst parameter has been a much-studied topic in various fields. In this section, we describe five simple estimators. Discussion of other methods for estimating the Hurst parameter will be postponed until the next section after we introduce random walk formulation and structure-function–based multifractal analysis.

1. **Variance-time plot**. With this method, one checks if Eq. (8.5) holds. If it does, then in a log-log plot of $var(X^{(m)})$ vs. the aggregate block size m, the curve should be linear for large m with slope larger than -1. By contrast, an SRD process has $var(X^{(m)}) \sim m^{-1}$, so that in a log-log plot of $var(X^{(m)})$ vs. block size m, the slope is -1. This is perhaps the most widely used method in traffic engineering.

2. **R/S statistic**: For a given set of observations $\{X_k, k = 1, 2, \cdots, n\}$ with sample mean $\overline{X}(n)$ and sample variance $S(n)^2$, the R/S statistic is given by

$$\frac{R(n)}{S(n)} = \frac{1}{S(n)}\left[\max(0, W_1, W_2, \cdots, W_n) - \min(0, W_1, W_2, \cdots, W_n)\right],$$

where

$$W_k = \sum_{i=1}^{k}[X_i - \overline{X}(n)], \tag{8.9}$$

and the factor $S(n)$ is introduced for the normalization purpose. As will be explained in Sec. 8.3, Eq. (8.9) defines a random walk. Therefore, $R(n)/S(n)$ essentially characterizes the normalized extent or range of the process W_k. One expects that the square of this extent scales with n as n^{2H}, similar to the scaling between the variance of a random walk and n. Indeed, we have

$$E\left[\frac{R(n)}{S(n)}\right] \sim n^H, \quad \text{as} \quad n \to \infty.$$

With small datasets, this method works better than the variance–time plot, since now one is required to compute the mean instead of the variance. Perhaps for this reason, the method is more popular in physiology. There, typically the datasets are small.

3. **Autocorrelation function–based estimator**: $r(k) \sim k^{-\beta}$ as $k \to \infty$. This estimator is, however, seldom used.

4. **Spectral density–based estimator**: As stated at the beginning of this section, the PSD for x is $S(f) \sim f^{1-2H}$ as $f \to 0$. This expression is very important in examining the consistency of different H estimators.

5. **Wavelet-based estimator**: In Sec. 4.2, we introduced wavelet MRA. Furthermore, in Sec. 6.5, we applied MRA to represent an fBm process. Eq. (6.32) is an effective method of estimating the Hurst parameter of an fBm process. If we start from a stationary time series, then we can expect that Eq. (6.32) will be modified to give a slope of $2H - 1$ instead of $2H + 1$, since the time series under study is like the fGn rather than the fBm. This is indeed the case, as shown by the following equations.

In Sec. 4.2, we find that wavelet MRA decomposes a signal $x(t)$ into

$$x(t) = \sum_k a_x(J,k)\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_k d_x(j,k)\psi_{j,k}(t).$$

Let

$$\Gamma_j = \frac{1}{n_j}\sum_{k=1}^{n_j} |d_x(j,k)|^2\,, \tag{8.10}$$

where n_j is the number of coefficients at level j; then the Hurst parameter is given by

$$\log_2 \Gamma_j = (2H-1)j + c_0, \tag{8.11}$$

where c_0 is some constant.

We note that either (1), (3), or (4) can be used to define the long-range dependence in a time series. Thus, methods (1), (3), and (4) actually constitute one independent estimator. Some researchers recommend using periodogram-based spectral estimation combined with Whittle's approximate maximum likelihood estimator to estimate the H parameter. With this approach, however, it may be difficult to determine a suitable region to define the power-law scaling.

8.3 RANDOM WALK REPRESENTATION AND STRUCTURE-FUNCTION–BASED MULTIFRACTAL ANALYSIS

In this section, we explain, first, how to construct a random walk process from a time series and, second, how to carry out multifractal analysis of the constructed random walk process based on the structure-function technique. The multifractal formulation will help us gain deeper understanding of the conventional methods for estimating the Hurst parameter and develop new means of estimating H.

8.3.1 Random walk representation

Let us denote the time series we want to study by $\{X_i, i = 1, 2, \cdots, N\}$. It can be any time series. For example, it can be a numerical sequence representing a

DNA sequence, or the interspike interval of neuronal firings, or switching times in ambiguous visual perceptions. For network traffic, it may be the interarrival time series, the packet length sequence, or the counting processes. For illustration purposes, in this section we use the counting process of a network traffic trace.

First, we subtract the mean from the time series. Denote the new time series as $\{x_i, i = 1, 2, \cdots, N\}$, where

$$x_i = X_i - \frac{1}{N}\sum_{j=1}^{N} X_j,$$

and consider it as a process similar to the fGn process. Then we form the partial summation of $\{x_i, i = 1, 2, \cdots\}$ to get the random walk process $\{y_k, k = 1, 2, \cdots\}$, where

$$y_k = \sum_{i=1}^{k} x_i. \tag{8.12}$$

Note that Eq. (8.12) is equivalent to Eq. (8.9). Figure 8.1 shows an example of the random walk process constructed from the counting process $\{\overline{B}_i, i = 1, 2, 3, \ldots\}$ of the network traffic trace we analyzed in Sec. 6.7. Even though the length of the random walk process shown in the figure is 2^{18} points long, it crosses a specific level, say $y = 0$, very rarely. This is a consequence of the fact that the H parameter for the process is much larger than 1/2. For this reason, levelset analysis of the process would require unrealistically long time series.

8.3.2 Structure-function–based multifractal analysis

After the random walk process is constructed, we can compute $Z^{(q)}(m)$ defined by

$$Z^{(q)}(m) = \langle |y(i+m) - y(i)|^q \rangle \sim m^{\zeta(q)}, \tag{8.13}$$

where the average is taken over all possible pairs of $(y(i+m), y(i))$, and examine whether the power-law scaling laws for different values of real q exist or not. Negative and positive q values emphasize small and large absolute increments of $y(n)$, respectively. Thus, the approach allows us to focus on different aspects of the data by using different q. In particular, when $q = 2$, the method is called fluctuation analysis (FA). When power-law scaling for some q exists, we say that the process under study is a fractal process. Furthermore, if $H(q)$, defined by

$$H(q) = \zeta(q)/q, \tag{8.14}$$

is not a constant function of q, we say that the process is multifractal. Note that when Eqs. (8.13) and (8.14) are combined, we can write

$$F^{(q)}(m) = \langle |y(i+m) - y(i)|^q \rangle^{1/q} \sim m^{H(q)}. \tag{8.15}$$

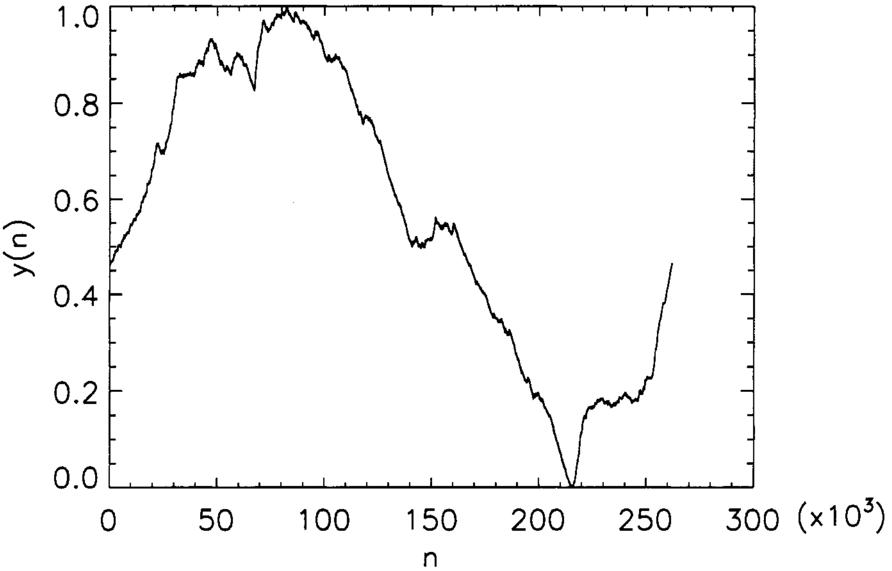

Figure 8.1. The random walk process constructed from the counting process of the network traffic trace, $\{\overline{B}_i, i = 1, 2, 3, \cdots\}$, analyzed in Sec. 6.7.

To make the idea concrete, Fig. 8.2 shows the result of multifractal analysis of the random walk process shown in Fig. 8.1. We observe that overall the curves are fairly linear; thus, the dataset can be classified as a fractal. However, if we look more carefully, we find that there is a knee point near $m = 2^{12}$. This means that there are two scaling regimes, $m \in (1, 2^{12})$ and $m \in (2^{12}, 2^{17})$. The $H(q)$ curves estimated for these two scaling regimes are shown in Fig. 8.3. We observe that for the scaling region $m \in (1, 2^{12})$, $H(q)$ slightly decreases with q, whereas for the scaling region $m \in (2^{12}, 2^{17})$, $H(q)$ decreases with q quite significantly. Thus, we conclude that for this specific random walk process, for scaling region $m \in (1, 2^{12})$, the process is a weak multifractal (i.e., more like a monofractal), while for the scaling region $m \in (2^{12}, 2^{17})$, the process is a multifractal. Since the scaling regions $m \in (1, 2^{12})$ and $m \in (2^{12}, 2^{17})$ correspond to short and long time scales, respectively, we can say that for this traffic trace data, it is almost a monofractal for short time scales and a multifractal for long time scales. This conclusion, however, may not be universal. In other words, some other traffic trace data may be multifractals for short time scales and monofractals for long time scales. It is also possible that some traffic data may be different types of mono– and/or multifractals on different time scales. We term this behavior nonuniversal fractal scaling for network traffic.

8.3.3 Understanding the Hurst parameter through multifractal analysis

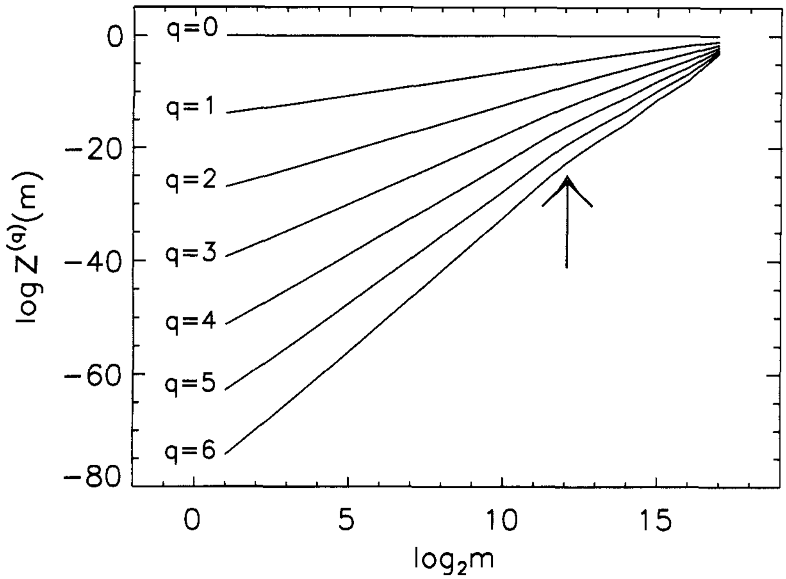

Figure 8.2. $Z^{(q)}(m)$ vs. m (in log-log scale) for the random walk process of Fig. 8.1.

In this subsection, we consider the meaning of the $H(q)$ spectrum from the multifractal analysis to develop new means of estimating the Hurst parameter. First, we note that $H(2)$ is simply the Hurst parameter H. To see this, we note that when $q = 2$, Eq. (8.13) reduces to the variance-time relation described by Eq. (8.5) if one notices that $\langle |y(i+m) - y(i)|^2 \rangle = m^2 var(X^{(m)})$.

Closely related to the variance-time relation is Fano factor analysis, which is quite popular in neuroscience. In the context of analysis of the interspike interval of neuronal firings, the Fano factor is defined as

$$F(T) = \frac{Var[N_i(T)]}{Mean[N_i(T)]}, \tag{8.16}$$

where $N_i(T)$ is the number of spikes in the ith window of duration T. For a Poisson process, $F(T)$ is 1, independent of T. For a fractal process, $Var[N_i(T)] \propto T^{2H}$ and $Mean[N_i(T)] \propto T$. Therefore, $F(T) \sim T^{2H-1}$. In other words, the Fano factor can be viewed as the relation between $[\langle |y(i+m) - y(i)|^2 \rangle / m]$ and m instead of the relation between $[\langle |y(i+m) - y(i)|^2 \rangle]$ and m .

We now discuss methods that employ $H(1)$ to estimate H. Two such approaches are reviewed by Taqqu et al. [424], namely, the Absolute Values of the Aggregated Series Approach and Higuchi's method. In the former, one determines if the following scaling law holds:

$$\frac{1}{[N/m]} \sum_{k=1}^{[N/m]} \left| X^{(m)}(k) \right| \sim m^{H-1},$$

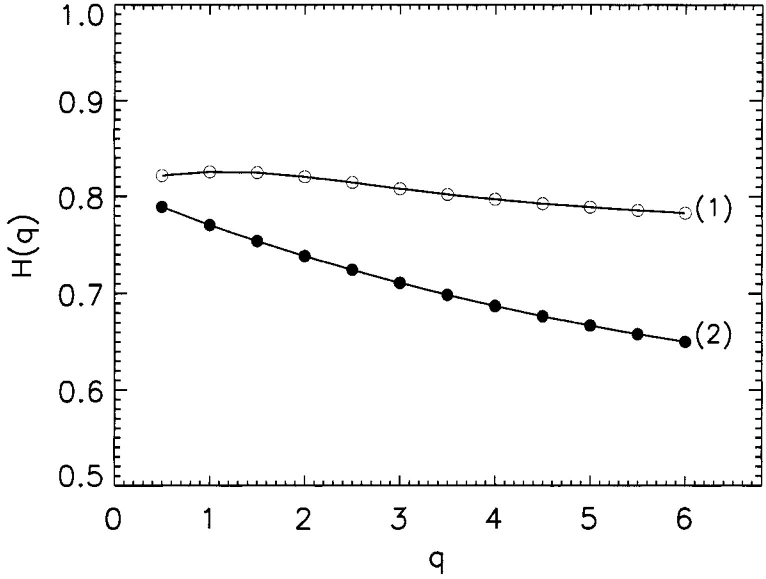

Figure 8.3. The variation of $H(q)$ vs. q computed from Fig. 8.2. Curves (1) and (2) represent $H(q)$ curves for two different scaling regimes, $m \in (1, 2^{12})$ and $m \in (2^{12}, 2^{17})$.

where N is the length of the time series, $X^{(m)}$ is the nonoverlapping running mean of X of block size m, as defined by Eq. (8.3), and [ ] denotes the greatest integer function. Higuchi's method, on the other hand, examines if the following scaling law is true:

$$L(m) = \frac{N-1}{m^3} \sum_{i=1}^{m} \left[(N-i)/m\right]^{-1} \sum_{k=1}^{[(N-i)/m]} \left| y(i+km) - y(i+(k-1)m) \right| \sim m^{H-2},$$

where N again is the length of the time series, m is essentially a block size, and $y(i)$ is the random walk process constructed from the X instead of the x time series. Note that the two methods are quite similar. In fact, the first summation of Higuchi's method divided by m is equivalent to the Absolute Values of the Aggregated Series Approach. The second summation, $\sum_{i=1}^{m}$, is another moving average, equivalent to taking overlapping running means of the original time series X. By now, it should be clear that both methods estimate the $H(1)$ parameter instead of the $H(2)$ parameter when the time series X has mean zero. The reason $H(1)$ can be used to estimate $H(2)$ is that typically they are quite close, even if the time series X is a multifractal. When the time series X is very much like a monofractal, or is only weakly multifractal, then we see that any $H(q)$, $q \neq 2$, can be used to estimate $H(2)$. In this case, the structure-function–based technique provides infinitely many ways of estimating the Hurst parameter.

We note that if the mean of the time series X is not zero, then neither the Absolute Values of the Aggregated Series Approach nor Higuchi's method estimates $H(1)$. When this is the case, one should remove the mean from the X time series first.

We have pointed out that Higuchi's method is equivalent to taking overlapping running means when constructing $X^{(m)}$. We thus see that the nonoverlapping condition for constructing $X^{(m)}$ when defining longmemory is not essential.

8.4 OTHER RANDOM WALK–BASED SCALING PARAMETER ESTIMATION

To illustrate further use of the random walk representation, we present two more ways of estimating the scaling parameter.

1. **Diffusion entropy analysis (DEA)**: Assume that a random walk process defined in a time interval $[t_0 + t, t_0]$ depends on the interval length t but not on the time origin. Based on the general definition of Eq. (6.1) for self-similar processes, Scafetta and Grigolini have found that the Shannon entropy for such a random walk process,

$$S(t) = -\int_{-\infty}^{\infty} dx\, p(x,t) \ln[p(x,t)], \tag{8.17}$$

where $p(x,t)$ is the PDF for the random walk, increases with t logarithmically,

$$S(t) \sim \delta \ln t, \tag{8.18}$$

where δ equals the Hurst parameter for fBm and Levy walk processes and equals the α parameter for Levy flight processes. To estimate $S(t)$, one can partition a random walk using a maximal overlapping window and then estimate $p(x,t)$ based on all the segments of the random walk process. While computationally the method is simple, it has a drawback: for a given time series of N points, the scaling behavior can be resolved at most up to $t = N/100$. This is because estimation of $p(x,t)$ for a random walk requires many (say, 100) sample realizations of the random walk.

2. **Phase space–based methods**: At the end of Sec. 6.4, we briefly mentioned that the Hurst parameter can also be estimated by monitoring the divergence of two nearby trajectories in a reconstructed phase space. The ideas will be easier to understand after we explain related concepts in Chapters 13 to 15. Therefore, we shall postpone this discussion until Chapters 14 and 15.

8.5 OTHER FORMULATIONS OF MULTIFRACTAL ANALYSIS

There are other formulations of multifractal, such as those based on detrended fluctuation analysis (DFA) and wavelet analysis. Let us first discuss the latter.

In Sec. 8.2, we discussed a wavelet-based method for estimating the Hurst parameter. The simple generalization of FA to the structure-function–based multifractal formulation discussed in Sec. 8.3 suggests that we generalize Eq. (8.10) to consider

$$\gamma_j(q) = \left[\frac{1}{n_j}\sum_{k=1}^{n_j} |d_x(j,k)|^q\right]^{1/q} \tag{8.19}$$

and examine whether the following scaling relations hold or not:

$$\gamma_j(q) \sim 2^{j[H(q)-1/2]}. \tag{8.20}$$

If they do, then we have

$$\log_2 \gamma_j(q) = j[H(q) - 1/2] + c_q, \tag{8.21}$$

where each c_q is some constant. This generalization is straightforward.

Next, we consider the DFA-based multifractal formulation. We first describe DFA. It is developed to first remove linear or other trends and then perform scaling analysis. Linear or other trends often exist in experimental data obtained under conditions that may be nonstationary.

DFA works as follows: First, divide a given random walk of length N into $\lfloor N/l \rfloor$ nonoverlapping segments (where the notation $\lfloor x \rfloor$ denotes the largest integer that is not greater than x); then define the local trend in each segment to be the ordinate of a linear least-squares fit for the random walk in that segment; finally, compute the "detrended walk," denoted by $y_l(n)$, as the difference between the original walk $y(n)$ and the local trend. Then one examines

$$F_d(l) = \left\langle \sum_{i=1}^{l} y_l(i)^2 \right\rangle^{1/2} \sim l^H, \tag{8.22}$$

where the angle brackets denote the ensemble average of all the segments and $F_d(l)$ is the average variance over all segments.

The extension of DFA to a multifractal formulation is also straightforward. Depending on how the deviation from a straight line in each window is characterized, one can have (at least) two forms. One is given by

$$F_d^{(q)}(l) = \left\langle \sum_{i=1}^{l} |y_l(i)|^q \right\rangle^{1/q} \sim l^{H(q)}, \tag{8.23}$$

where q is real, taking on both negative and positive values. Another is given by

$$F_d^{(q)}(l) = \left\langle \left[\sum_{i=1}^{l} |y_l(i)|^2\right]^{q/2} \right\rangle^{1/q} \sim l^{H(q)}. \tag{8.24}$$

The first formulation amounts to using the l_1 norm, while the second uses the l_2 or Euclidean norm.

8.6 THE NOTION OF FINITE SCALING AND CONSISTENCY OF H ESTIMATORS

To facilitate our discussion on the correlation structure of ON/OFF intermittency and Levy motions in this section, as well as numerous applications in Sec. 8.8, we consider two important issues. One is finite scaling behavior. The other is consistency of the estimated H using different estimators. Let us illustrate both issues with simple examples.

Imagine two people playing a simple game: Hanna tosses a fair coin, and Albert guesses whether the outcome is a head or tail and compares his guess with the actual outcome. Let head and tail be denoted by 1 and -1, respectively. Occasionally, they observe a long sequence of heads or tails. Being an intelligent adult, Albert reasons that whenever a sequence of heads (or tails) appears, it would be better to guess the next one to be a tail (or head), since the probability for a head (or tail) to appear after a sequence (say, $n = 4$) of heads (or tails) has occurred is very low. Intuitively, one expects that a sequence of 1 and -1 guessed by Albert yields $H < 1/2$, i.e., antipersistence, for a small scale but $H = 1/2$ for not too small a scale. This is indeed so, as can be readily verified by carrying out such an experiment. To save time, however, let us use a computer and perform a simple simulation.

First, we generate a sequence of 1 and -1 equivalent to tossing a fair coin. Next, we sequentially search the sequence to check whether a sequence of 1 (or -1) has occurred with length greater than a parameter n. Whenever we find such a patch of 1 (or -1), we replace the nth 1 (or -1) of that sequence by -1 (or 1), and then resume our search, starting from the position that has just been modified, until the end of the sequence. The resulting sequence would be equivalent to Albert's sequence of 1 and -1. Figure 8.4 shows $\log_2 F(m)$ vs. $\log_2 m$ for $n = 3$ (denoted by the circle) and 4 (denoted by the asterisk), respectively. Clearly we observe that $H < 1/2$ for small m but is equal to 1/2 when m is not too small. This is an excellent example showing the meaning of both antipersistence and finite scaling (or multiple fractal scaling).

Let us now consider time series generated from an AR(1) model, $x_{n+1} - \overline{x} = a(x_n - \overline{x}) + \eta_n$, discussed in Chapter 3. For this model, the autocorrelation decays exponentially: $C(m) = \sigma^2 a^{|m|}$ (Eq. (3.36)). When the time lag m is large, the correlation is essentially zero; we can expect H to be 1/2. However, when the coefficient a is only slightly smaller than 1, $C(m)$ will be close to 1 for a considerable range of m. In this case, we have almost perfect correlation. One thus might expect $H \approx 1$ for not too large a time scale. This seems to be verified when one applies the variance-time relation to analyze the generated time series or, equivalently, applies FA to the random walk process constructed from the data. The latter is shown in Fig. 8.5(a). However, there is a problem here: If we employ DFA, then we obtain $H = 1.5$ for not too large a time scale, as shown in Fig. 8.5(b). What is going on?

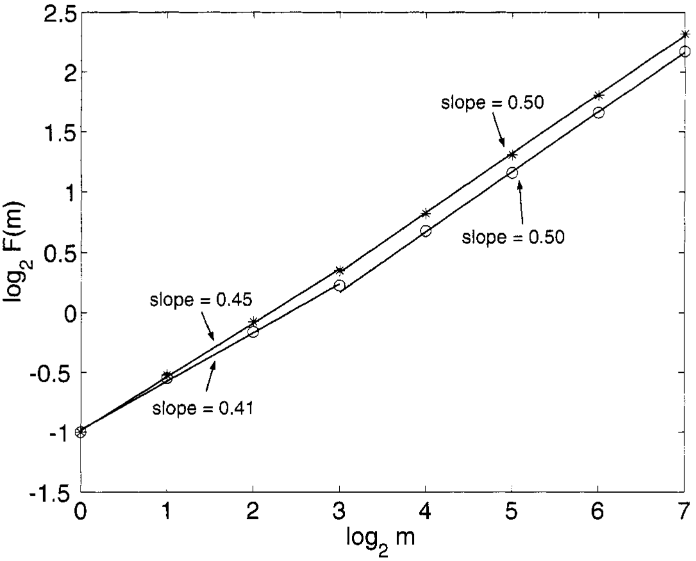

Figure 8.4. Antipersistence in guessing the outcome of a coin toss.

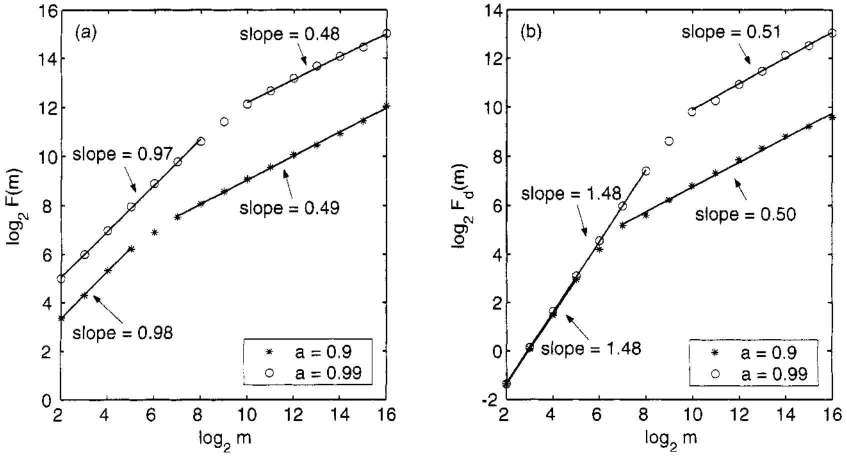

Figure 8.5. H parameter for the AR(1) model.

Which result should be trusted? To find the answer, let us examine which one is consistent with the PSD of the AR(1) process.

As we discussed in Sec. 4.12, the PSD of an AR(1) process is

$$S_x(f) = \frac{\sigma_n^2}{1 + a^2 - 2a\cos\omega}, \quad 0 \le \omega \le \pi, \tag{8.25}$$

where σ_n^2 is the variance of the noise. Expanding $\cos\omega = 1 - \omega^2/2 + \cdots$ and noting that $\omega = 2\pi f$, we have

$$S_x(f) \approx \frac{\sigma_n^2}{(1-a)^2 + a(2\pi)^2 f^2}, \quad 0 \le f \le 1/2.$$

At the low-frequency end, the term $a(2\pi)^2 f^2$ can be dropped, and we have a flat spectrum consistent with $H = 1/2$. At the high-frequency ($f \to 1/2$) end, since a is close to 1, the term $(1-a)^2$ can be dropped, and we have

$$S_x(f) \propto f^{-2}.$$

The transition frequency f_* is found by equating the two terms:

$$(1-a)^2 \approx a(2\pi)^2 f_*^2.$$

From this, we get

$$T^* = 1/f_* = \frac{2\pi\sqrt{a}}{1-a}.$$

To find more precisely the frequency ranges where the PSD is flat or decays as f^{-2}, we may require $(1-a)^2 \gg a(2\pi)^2 f^2$ when $f \le f_1$ and $(1-a)^2 \ll a(2\pi)^2 f^2$ when $f_2 \le f \le 1/2$. Quantitatively, $f_{1,2}$ may be defined by the following conditions:

$$(1-a)^2 = \theta a(2\pi)^2 f_1^2$$

and

$$\theta(1-a)^2 = a(2\pi)^2 f_2^2,$$

where the parameter θ is on the order of 10. The two time scales defined by $f_{1,2}$ are $T_{1,2} = 1/f_{1,2}$, with

$$T_1 = T^*/\sqrt{\theta}, \quad T_2 = T^*\sqrt{\theta}.$$

Let us now examine Fig. 8.5 again. From either FA or DFA plots, we indeed observe that around T^*, the scaling changes from a large H (1 for FA and 1.5 for DFA) to $H = 1/2$. For $1/f^\beta$ noise, $\beta = 2H - 1$. Now that $\beta = 2$, we have to conclude that $H = 1.5$. Therefore, DFA is consistent with the spectrum, but FA is not!

At this point, it is important to note that the AR(1) model with coefficient a very close to 1 has been proposed as a (pseudo) model for LRD traffic with $H = 1$ and a convenient model for exact $1/f$ noise. The former mis-interpretation is due to misuse of FA (or the variance-time relation) on the data; the latter may occur because the magnitude response of the Fourier transform of the process, $|X(\omega)|$, scales with f as f^{-1} when $f \to 1/2$. When $|X(\omega)|$ is mistaken for PSD, one would claim that the model generates the exact $1/f$ spectrum.

Now let us understand why FA gives $H = 1$ for the AR(1) model. We approach the problem from a broader perspective by asking this question: Given a measured time series $x_1, x_2, \cdots$, shall we treat it as a noise process like fGn or as a random walk process like fBm? If the data show a noise process, then we have to form a partial summation according to Eq. (8.12) if we wish to apply the structure-function–based multifractal analysis. On the other hand, if the data show a random walk process, then no partial summation is needed when applying the structure-function–based multifractal analysis. However, one has to obtain the noise process by differencing the original data when applying the variance-time relation, Fano factor analysis, or the R/S statistic.

To answer the question posed above, let us be more quantitative. Denote an ideal fractal process by $x_1, x_2, \cdots$. Let the process obtained by removing the mean of x and integrating it be denoted by the y process. If the PSD for the x time series is $1/f^{\beta_x}$, then the PSD for y is $1/f^{\beta_y}$, where $\beta_y = \beta_x + 2$. As we pointed out earlier, $\beta_x = 2H_x - 1$ and $\beta_y = 2H_x + 1$ (where the subscript x is used to emphasize that this H is associated with the x process). Here it appears that we are all right, since the two equations appear to be equivalent (but there is a hidden complexity, as we shall point out soon).

Now let us start from the y process and integrate it to obtain the z process. The PSD for z is $1/f^{\beta_z}$, with $\beta_z = \beta_y + 2$. We expect to get $\beta_y = 2H_y - 1$ and $\beta_z = 2H_y + 1$. Now what is the relation between H_x and H_y? It is simple and is given by

$$H_y = H_x + 1. \tag{8.26}$$

It seems that everything is still all right. By this argument, one expects H to increase by 1 each time the process is integrated. So where is the complexity?

The trouble lies in the simple fact that our theory demands $0 < H < 1$. As we shall show shortly, FA and other equivalent methods for estimating H only return an H that belongs to the unit interval, no matter what types of data are analyzed. For example, if we estimate H_y by the variance-time relation, or Fano factor analysis, or the R/S statistic, or FA, the value of H_y will be at most 1! However, if $0 < H_x < 1$, then by Eq. (8.26), we should have $1 < H_y < 2$. This is the difficulty!

To understand why H estimated by FA or other equivalent methods saturates at 1, let us assume that $y(n) \sim n^\gamma$, $\gamma > 1$. Then $\langle |y(n+m) - y(n)|^2 \rangle = \langle [(n+m)^\gamma - n^\gamma]^2 \rangle$ is dominated by terms with large n. When this is the case, $(n+m)^\gamma = [n(1+m/n)]^\gamma \approx n^\gamma[1+\gamma m/n]$. One then sees that $\langle |y(n+m) - y(n)|^2 \rangle \sim m^2$, i.e., $H(2) = 1$. Similarly, one can prove that the smallest H given by FA is 0.

By now, it should be clear why FA gives $H = 1$ for the high-frequency end of an AR(1) process, while the PSD and DFA give $H = 1.5$. It turns out that DFA saturates at $H = 2$ from above and $H = 0$ from below. Wavelet analysis is the most versatile; H can be both negative and larger than 2, and multiple integrations or differentiations of a measured time series are all fine. However, for practical purposes, DFA can be considered adequate.

For ease of practical applications, we list a few rules of thumb:

Rule of thumb 1: When a time series is treated as a noise process and the estimated H parameter is close to 1, question your result; redo the analysis by treating the data as a random walk process.
Rule of thumb 2: When a time series is treated as a random walk process and the estimated H parameter is close to 0, do not trust your result; redo the analysis by integrating the data.
Rule of thumb 3: To be safe, perform DFA or the wavelet-based analysis on your data, along with FA (or other equivalent methods), and check the consistency of the results based on different methods.

8.7 CORRELATION STRUCTURE OF ON/OFF INTERMITTENCY AND LEVY MOTIONS

ON/OFF intermittency is a ubiquitous and important phenomenon. For example, a queuing system or a network can alternate between idle and busy periods; a fluid flow can switch from a turbulent motion to a regular (called laminar) one. In this section, we study the correlation structure of an ON/OFF train. It turns out that the correlation structures of ON/OFF models and Levy walks are closely related. Therefore, we shall also study the latter in this section. At the end of this section, the distinction between FA and DFA will be further illustrated.

8.7.1 Correlation structure of ON/OFF intermittency

Let us denote an ON period by 1 and an OFF period by 0. We study three types of ON/OFF trains where ON and OFF periods are independent and both have the same (1) exponential distribution, (2) Pareto distribution (Eq. (3.24)), and (3) Pareto distribution with truncation. For Pareto distributions, we choose two α: 1.6 and 0.6. Truncation is achieved by simply requiring $x \le L$, where L is a parameter. When $1 \le \alpha \le 2$, it can be proven that

$$H = (3 - \alpha)/2. \qquad (8.27)$$

One of our purposes is to check whether Eq. (8.27) can be numerically verified. To do this, we apply FA and DFA to the integrated data of an ON/OFF train. The ON/OFF train is sampled in such a way that in a total of about 1000 ON/OFF periods, on average a few tens of points of an ON or OFF period are sampled. The results for FA and DFA are shown in Figs. 8.6(a,c) and (b,d), respectively. We observe that for these three cases, for a small time scale (determined by the average length of an ON or OFF period), H (as the slopes of the lines) is close to 1 by FA and 1.5 by DFA. By simple analytical analysis or numerical simulation, one can readily find that for high frequency, the PSD for an ON/OFF train scales with the frequency as f^{-2}, just like the high-frequency end of an AR(1) model. Therefore, for time scales not longer than the average ON or OFF period, DFA is consistent

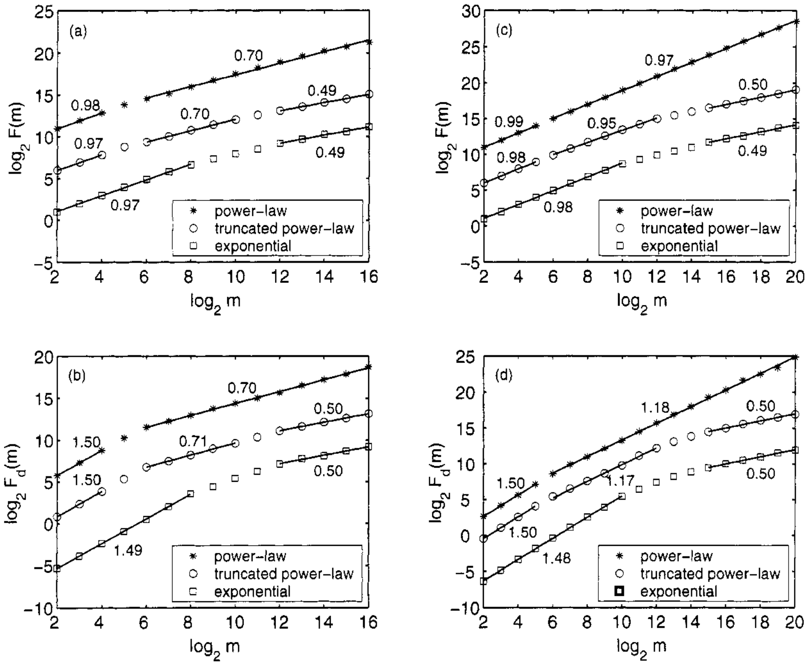

Figure 8.6. H parameter for the ON/OFF model. The α parameter is 1.6 for (a,b) and 0.6 for (c,d).

with PSD but FA is not. For larger scales, for case (1), we observe that H from both FA and DFA is 0.5 (with regard to Eq. (8.27), this amounts to taking $\alpha = 2$), while for cases (2) and (3), we observe that Eq. (8.27) is correct with FA when $1 \leq \alpha \leq 2$ and correct with DFA for the entire range of admissible α: $0 \leq \alpha \leq 2$. When $0 \leq \alpha < 1$ due to saturation, FA always gives $H = 1$. When the power-law distribution is truncated, H eventually becomes 1/2, both by FA and DFA.

8.7.2 Correlation structure of Levy motions

In Chapter 7, we studied stable laws and Levy motions. A stable law is a distribution with heavy tails. There are two types of Levy motions. One is Levy flights, which are random processes consisting of many independent steps, each step characterized by a stable law. The other is Levy walks, where the time consumed on each step is proportional to its length. As we noted in Chapter 7, a Levy walk can be obtained by resampling a Levy flight. Intuitively, we expect Levy flights to be memoryless, simply characterized by $H = 1/2$, irrespective of the value of the exponent α characterizing the stable laws. This is indeed the case, as is shown in Fig. 8.7. The correlation structure of a Levy walk, however, is more complicated. We observe from Fig. 8.8 that when the scale is small, corresponding to "walking" along a single step of a Levy flight, H is close to 1 by FA and close to 1.5 by DFA. Analysis by

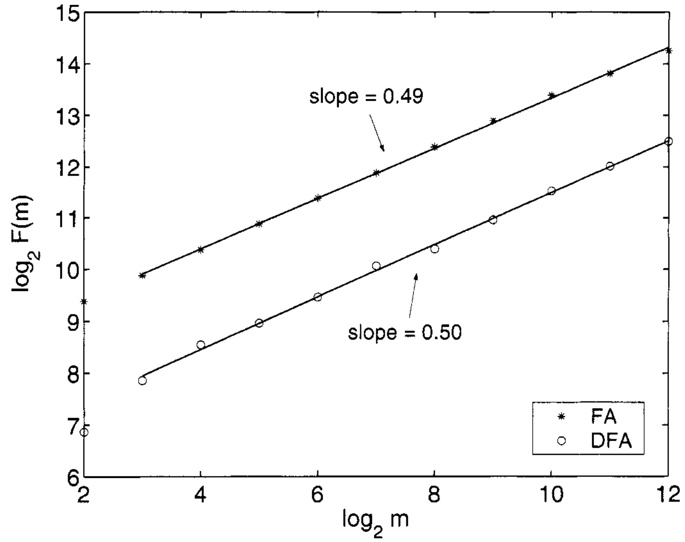

Figure 8.7. *H* parameter for Levy flights. *H* is independent of the parameter α.

Fourier transform shows that the PSD at the high-frequency end again decays as f^{-2}. Therefore, on smaller scales, DFA is consistent with PSD but FA is not. On larger scales, corresponding to constantly "switching" from one step of a Levy flight to another, H is given by Eq. (8.27) for FA when $1 \leq \alpha \leq 2$ and for DFA when $0 \leq \alpha \leq 2$. Again due to saturation, FA always yields $H = 1$ when $0 \leq \alpha < 1$. While these observations are similar to those found for the ON/OFF trains discussed above, we note a difference between Figs. 8.6 and 8.8. That is, for a Levy walk, the transition from a larger H at a small scale to a smaller H at a large scale is more gradual. This difference is due to the difference between a stable law and a Pareto distribution.

To further illustrate the similarity between Levy walks and ON/OFF trains, we have shown in Fig. 8.9 the data obtained by differencing a Levy walk. They resemble an ON/OFF train. Indeed, a moment's thought should convince one that when walking within a step of a Levy flight, one will be on one of the ON (or OFF) levels. However, depending on the stepsize of the Levy walk, around the transitions from one step of the original Levy flight to another, the pattern may deviate slightly from the ON/OFF train.

8.8 DIMENSION REDUCTION OF FRACTAL PROCESSES USING PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a method for reducing the dimension of the original data by projecting the raw data onto a few dominant eigenvectors with large

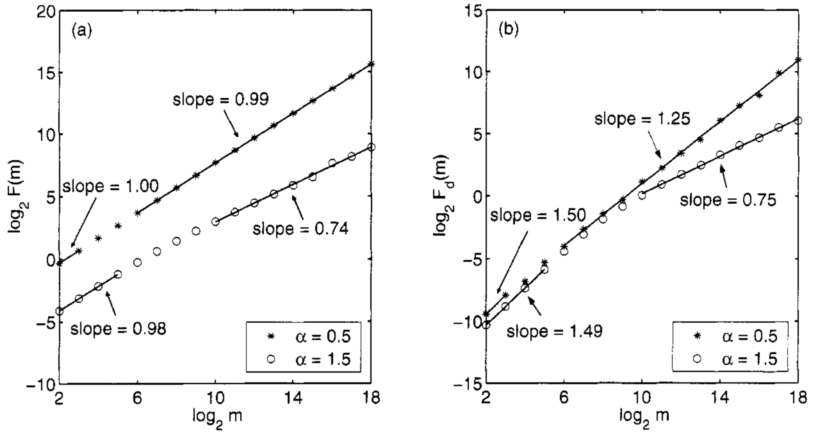

Figure 8.8. *H* parameter for Levy walks.

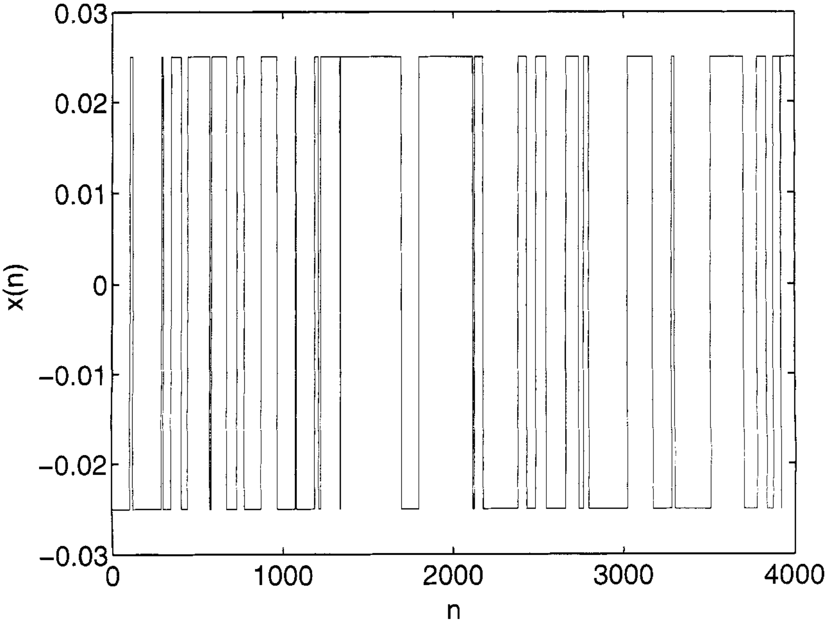

Figure 8.9. Difference data for a Levy walk.

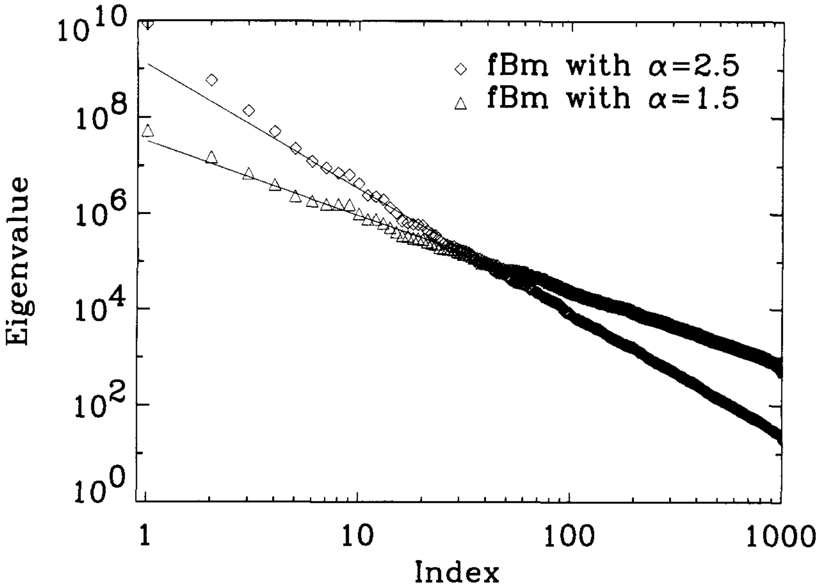

Figure 8.10. Eigenvalue spectrum for the fBm processes with $H = 0.25$ and 0.75.

variance (energy). It is closely related to singular value decomposition (SVD). In the continuous case, it is called Karhunen-Loève (KL) expansion. The latter is often called proper orthogonal decomposition (POD) in turbulence and empirical orthogonal functions (EOFs) in meteorology. In Appendix B, we have provided a brief description of PCA, SVD, and KL expansion.

Because of its conceptual simplicity and widely available codes based on well-studied numerical schemes, PCA has recently been used to analyze DNA microarray data and brain functional magnetic resonance imaging (fMRI) data. To reduce the dimension of some measured data, it is often assumed that the raw data may be projected onto a few dominant eigenvectors with large variance (or energy). The ubiquity of fractal processes compels us to examine whether PCA can be used to reduce the dimension of these processes.

The above question can be readily answered by performing PCA on a fractal time series. Let us consider fBm processes first. It turns out that the rank-ordered eigenvalue spectrum for a fractal process with the Hurst parameter H decays with the index i as a power law, with the exponent being $2H + 1$. An example is shown in Fig. 8.10.

Next, we consider PCA of a random walk constructed from the DNA sequence of the bacteria lambda phage. In Sec. 4.1.5, example 3, we explained how a DNA walk can be constructed. For the DNA walk of the bacteria lambda phage shown in Fig. 8.11(a), the rank-ordered eigenvalue spectrum is shown in Fig. 8.11(b). Again we observe a power-law decaying eigenvalue spectrum with exponent $\beta = 2.034$.

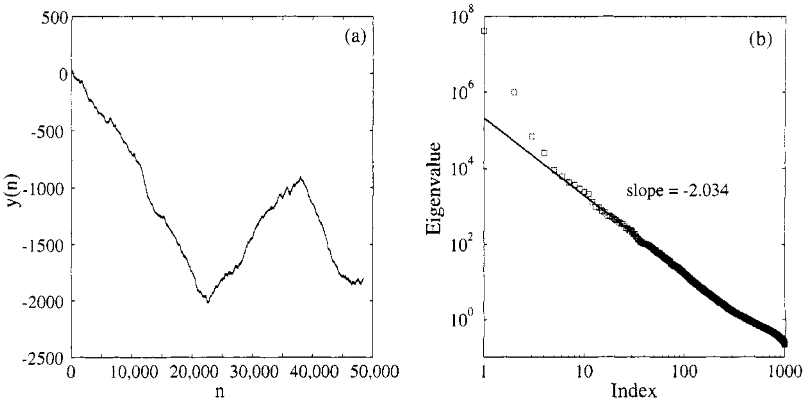

Figure 8.11. The random walk for the DNA sequence of the bacteria lambda phage (a) and its eigenvalue spectrum (b).

The Hurst parameter is then $H = (\beta - 1)/2 = 0.517$, very close to the value estimated using DFA, which is 0.51.

It is interesting to note that the bacteria lambda phage DNA has long patches. FA of its DNA walk yields $H \approx 0.56$, leading to the interpretation that this DNA sequence has longmemory. The original motivation for developing DFA was to remove this patchy effect. DFA achieves this goal. PCA can serve the same purpose (except that it is much slower than DFA).

The analysis of the above two examples is easy to interpret: fractal time series cannot be represented by a few dominant eigenvectors and eigenvalues. Let us now perform a more complicated analysis — apply PCA to network data.

A network can be represented by an adjacency matrix$A$, where the elements a_{ij} are zeros when there is no interaction between the nodes$i$ and j. For an undirected network, a_{ij} can be assigned a value of 1 when node$i$ interacts with node j. This results in a symmetric matrix$A$ and can be readily analyzed by eigen-decomposition. For a directed network, a_{ij} can be assigned a value of either 1 or-1, depending on whether$i$ "controls" or is being "controlled" by$j$. The matrix A is then asymmetric. Note that the elements of the matrix may be assigned numbers other than± 1 or 0 to explicitly reflect coupling strength. This is the case for microarray data. To analyze such matrices with eigen-decomposition, one can form eitherAA^T or A^TA, where the superscript T denotes transpose, and perform PCA. Alternatively, one can simply perform SVD on A.

Recently, it has been found that a power-law eigenvalue spectrum can be generated by a power-law network such as the global Internet. When PCA is applied to molecular interaction network data, such as microarray data (which monitor the expression of thousands of genes simultaneously under various conditions) and protein-protein interaction data, a powerlaw-like eigenvalue spectrum can also be observed. A few examples are shown in Fig. 8.12, where the data of (a) were used

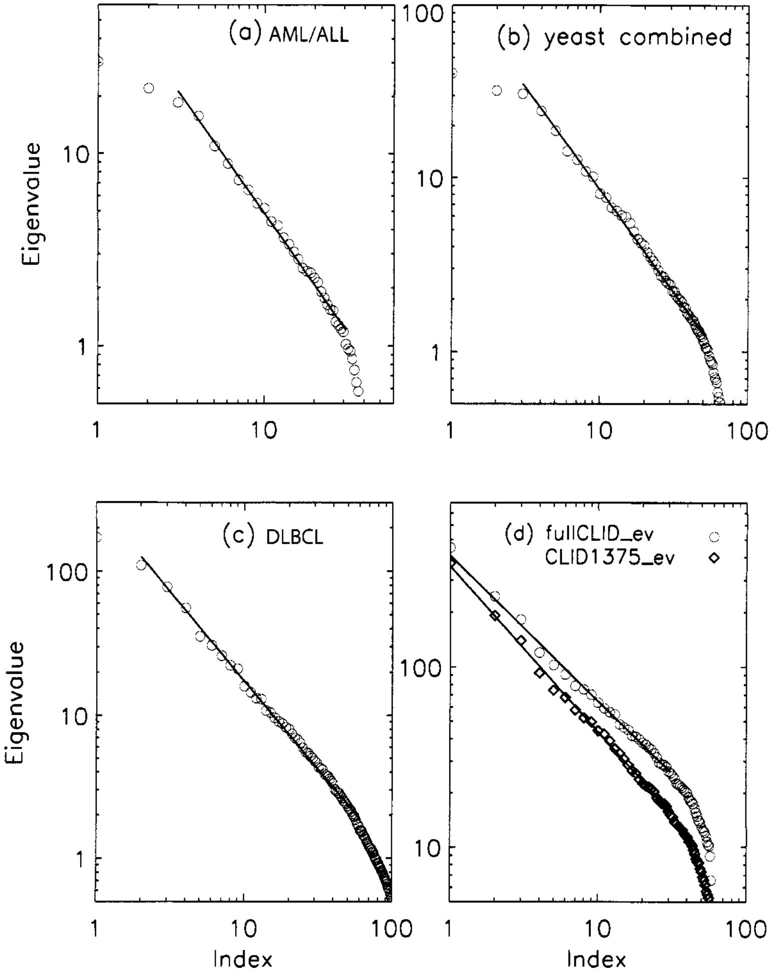

Figure 8.12. Eigenvalue spectrum, in log-log scale, for four microarray datasets.

to classify leukemia into acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL), the data of (b) were used to study sporulation in budding yeast, the data of (c) were used for the classification of diffuse large B-cell lymphoma (DLBCL), and the data of (d) were used to study systematic variation in gene expression patterns in 60 human cancer cell lines. There are two curves in (d): one is for the full dataset, the other for a subset of the dataset (which was considered to have higher quality than the rest of the data). The power-law distributed eigenvalue spectra observed here may be a signature of dynamic correlations among different pathways expressed by microarray data in a gene transcriptional network.

We should emphasize here that although a power-law eigenvalue spectrum implies that the data cannot be described by a few dominant eigenvectors and eigenvalues, when one's purpose is to classify data into a few groups, one can try to perform PCA and retain only a few eigenvalues and eigenvectors, so long as one's goal can be effectively achieved. This is different than data modeling. Of course, in this regard, it is possible that eigenvalues and eigenvectors other than the few dominant ones may serve one's purpose better.

At this point, we should also mention that the eigenvalue spectrum for a chaotic time series decays exponentially. However, the number of eigenvalues following the exponential decay is typically larger than the dimension of the chaotic attractor. In this case, for the purpose of modeling the chaotic data, one has to retain all the exponentially decaying eigenvalues and their corresponding eigenvectors. However, for the purpose of pattern classification, one can use a smaller number of eigenvalues and eigenvectors, as noted earlier.

8.9 BROAD APPLICATIONS

The materials discussed in this chapter have found numerous applications in fields as diverse as physics, geophysics, physiology, bioinformatics, neuroscience, finance, and traffic engineering, among many others. Analysis and modeling of traffic data has already been discussed in this chapter and in Chapter 6. Below we examine three more examples: target detection within sea clutter radar returns, deciphering the causal relation between neural inputs and movements by analyzing neuronal firings, and gene finding from DNA sequences. These examples will further illustrate the importance of the two notions discussed in Sec. 8.6: fractal scaling break and consistency of the H estimators. This way, our appreciation of the power as well as the limitations of the concepts and methodologies will be greater.

8.9.1 Detection of low observable targets within sea clutter

Sea clutter is the backscattered returns from a patch of the sea surface illuminated by a radar pulse. Robust detection of targets from sea clutter radar returns is an important problem in remote sensing and radar signal processing applications. This is a difficult problem because of the turbulent wave motions on the sea surface as well as multipath propagation of radar pulses. In the past several decades, great efforts have been made to understand the nature of sea clutter as well as to detect targets within sea clutter. However, novel, simple, and reliable methods for target detection are yet to be developed. In this subsection, we show that the $H(q)$ spectrum together with the finite fractal scaling range offer a very simple and effective method to detect low observable targets within sea clutter.

First, we note that the details of the sea clutter data can be found in Sec. A.2 of Appendix A. The following analysis is based on 392 sea clutter time series, each containing 2^{17} complex numbers, sampled with a frequency of 1000 Hz. Visually, all these time series are similar to the one shown in Figs. 1.4(a–d). Similar signals

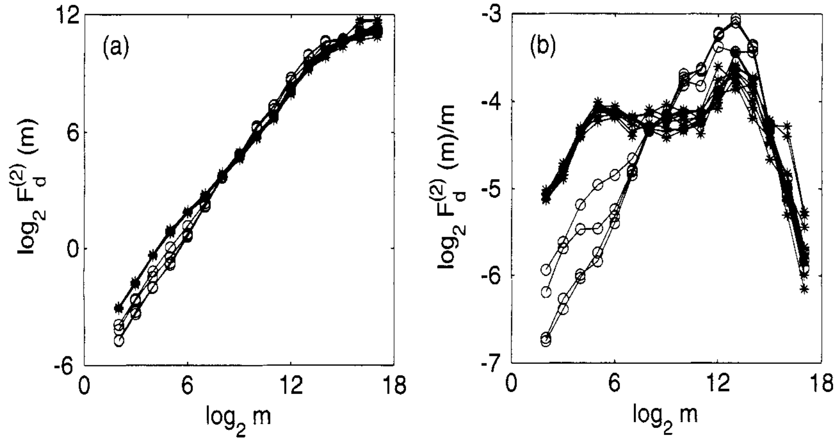

Figure 8.13. Target detection within sea clutter using DFA. Open circles designate data with a target, while asterisks designate data without a target.

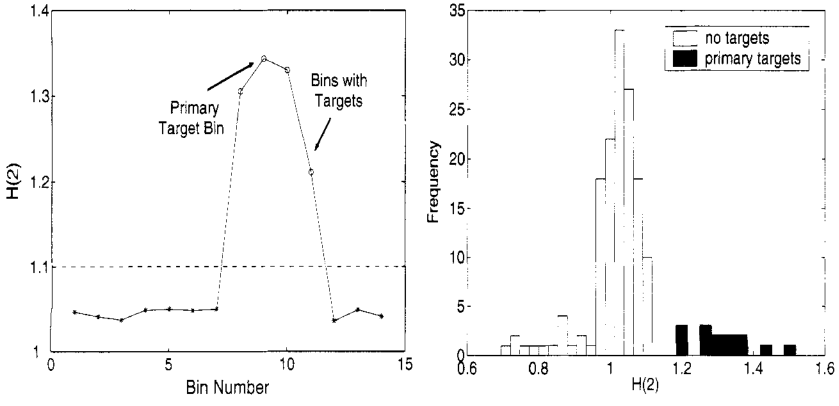

Figure 8.14. Left: $H(2)$ parameter for the 14 range bins of a measurement; Right: Histogram (equivalent to PDF) for the $H(2)$ parameter for all the measurements.

have been observed in many different fields. Therefore, the analysis below may also be applicable to those fields.

Let us denote the sea clutter amplitude data by $u_1, u_2, \cdots$, the integrated data by $v_1, v_2, \cdots$, and the differenced data by $w_1, w_2, \cdots$. First, we apply DFA to $v_1, v_2, \cdots$. A typical result for a measurement (which contains 14 range bins) is shown in Fig. 8.13(a). From it, one would conclude that the data have excellent fractal scaling behavior. However, this is an illusion due to the large y-axis range in the figure. If one reduces the y-axis range by plotting $\log_2[F_d^{(2)}(m)/m]$ vs. $\log_2 m$ (which can be viewed as detrended Fano factor analysis; see Sec. 8.3.3), then one finds that the curves for sea clutter data without a target change abruptly around

$m_1 = 2^4$ and $m_2 = 2^{12}$. Since the sampling frequency is 1000 Hz, they correspond to time scales of about 0.01 and 4 s. It turns out that if one fits a straight line to the $\log_2[F_d^{(2)}(m)/m]$ vs. $\log_2 m$ curves in this m range, then the H parameter can completely separate data with and without a target, as shown in Fig. 8.14. The last statement simply says that the H-based method achieves very high accuracy in detecting targets within sea clutter. We note that $H(q),\ q \neq 2$ is similarly effective in detecting targets. Below we shall focus on $H(2)$, and simply abbreviate it as H.

Let us now make a few comments: (1) The time scales of 0.01 s and a few seconds have specific physical meanings: below 0.01 s, the data are fairly smooth and hence cannot be fractal; above a few seconds, the wave pattern on the sea surface may change; hence, the data may have a different behavior (possibly another type of fractal). With the available length of the data (which is about 2 min), the latter cannot be resolved, however. (2) If one tries to estimate H from other intervals of time (which would be the case when one tries to apply, say, maximum likelihood estimation), then H fails to detect targets within sea clutter. (3) The fractal scaling in the identified time scale range is not well defined, especially for data without a target. This implies that sea clutter data are too complicated for fractal scaling to characterize. (4) If one applies DFA to the u_i process, the original sea clutter amplitude data, then the estimated H_u is about $H_v - 1$, and the H-based method for target detection still works. (5) When FA is applied to the u_i process, the obtained H are similar to those obtained by DFA. Hence, FA is consistent with DFA. However, FA fails to work when it is applied to the integrated data, the v_i process, since all the estimated H_v cannot be larger than 1. (6) The wavelet H estimator is the most versatile. The H values obtained by applying the method to the u_i and v_i processes as well as the w_i process can all be used to detect the target, as shown in Fig. 8.15. In fact, H is increased by 1, progressing from w_i to u_i and from u_i to v_i. Neither FA nor DFA gives useful results when applied to the w_i process because of saturation of H at 0. (7) H for some datasets with targets is close to $1/3$, the H corresponding to the famous Kolmogorov energy spectrum of turbulence. This may be due to the development of wave-turbulence interactions around the target under favorable weather and sea conditions.

At this point, we ask a question: Why should the H from wavelet analysis of the u_i process (Fig. 8.15(b)) be compared to the H from DFA of the v_i process (Fig. 8.14)? We leave this as exercise 5.

8.9.2 Deciphering the causal relation between neural inputs and movements by analyzing neuronal firings

Biological systems have many properties such as complexity, robustness, reliability, and degeneracy. Such properties arise from the interactions of the components of a system as well as from the features of those components. Even without deep understanding of the dependence of those properties on the structure and features

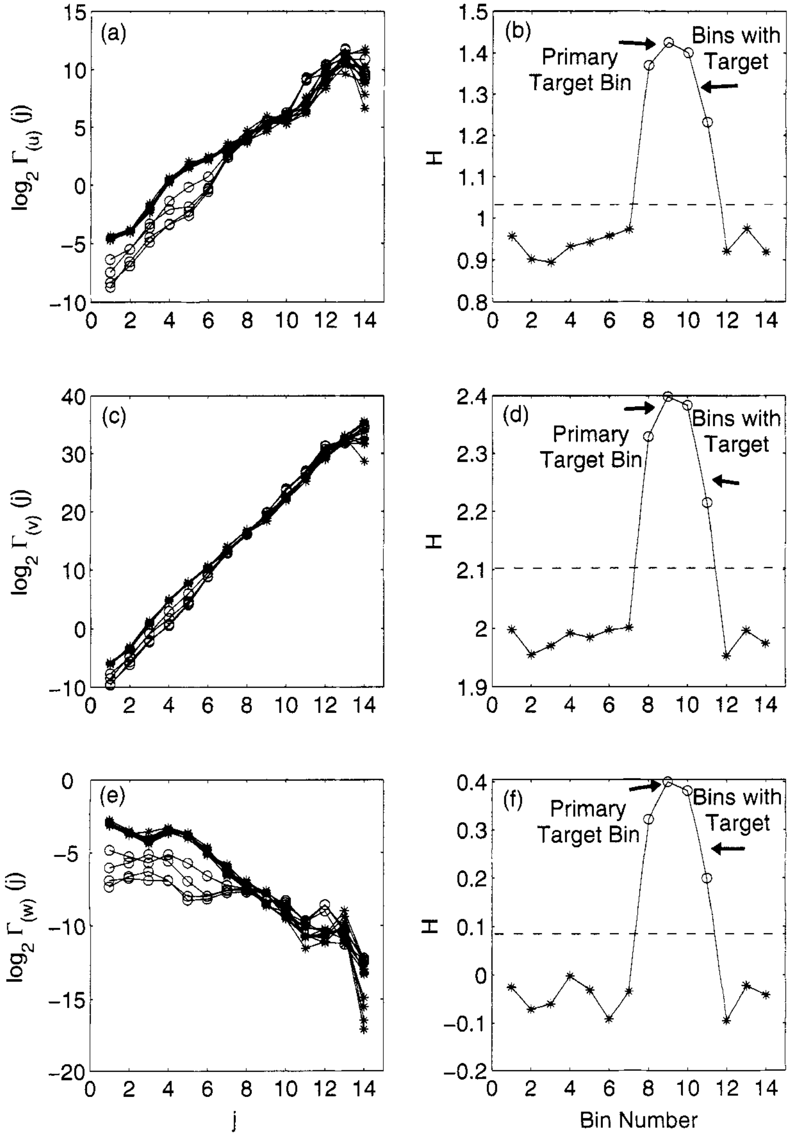

Figure 8.15. Wavelet H estimator on (a,b) the u_i process, (c,d) the v_i process, and (e,f) the w_i process.

of biological systems, those properties have been providing guidelines for the design of many artificial systems. One fascinating artificial system is the brain-machine interface (BMI), whose purpose is to provide a method for people with damaged sensory and motor functions to use their brain to control artificial devices and restore their lost ability via the devices.

In recent years, many BMI researchers have demonstrated the feasibility of using adaptive input-output models to map the fundamental timing relations between neural inputs and hand movement trajectory. To achieve the mapping, model parameters are chosen in such a way that the difference between model output and hand movements is minimized using a statistical criterion such as meansquare error. The adaptive models used so far usually contain a very large number of parameters and require extensive training. This limits the optimal correlation between model output and hand trajectory to around 70–80% and prevents researchers from gaining deep understanding of the causal relation between neural inputs and hand movements. Furthermore, adaptive models assume that neuronal firings in the cortex are stationary, while in fact they are not (as will be shown shortly). To fundamentally advance the state of the art of BMI research, it has become increasingly important to develop new theoretical frameworks and methods to better understand neural information processing through characterization of the spatial-temporal dynamics of the firing patterns of a population of neurons. In this subsection, we show that DFA can help reveal causal relations between neuronal firings and hand trajectory. For details of the data, see Sec. A.3 of Appendix A.

8.9.2.1 Varying degree of correlation between neuronal firings and hand trajectory To understand how neuronal firings control hand trajectory, it is instructive to examine the correlation between them. For this purpose, three consecutive hand movements are shown in Fig. 8.16(a), while neuronal firings of five neurons associated with those hand movements are shown in Figs. 8.16(b–f). A number of interesting features can be observed from Figs. 8.16(b–f):

1. The firing rate varies considerably among the neurons. For example, neuron 1, plotted in Fig. 8.16(b), fired far more than most other neurons.
2. The firing of neuron 1 in Fig. 8.16(b) does not have much correlation with hand trajectory. In fact, out of 104 neurons whose firings were recorded, more than half had little correlation with hand trajectory.
3. While the firing patterns of neurons 2–4, plotted in Figs. 8.16(c–e), appear to have strong correlations with the hand movement trajectory, the degree of correlation varies considerably with time. For example, neurons 2, 3, and 4 did not fire much during the monkey's first, second, and third periods of hand movement, respectively. It should be mentioned that although neuron 5, shown in Fig. 8.16(f), fired often during all three periods, it also had "quiet" periods even though the monkey was actively grabbing food and bringing it to its mouth.

These observations suggest that (1) different neurons have different degrees of importance in determining the causal relation between neural inputs and hand movements and that (2) even for the same neuron, the degree of importance varies considerably with time.

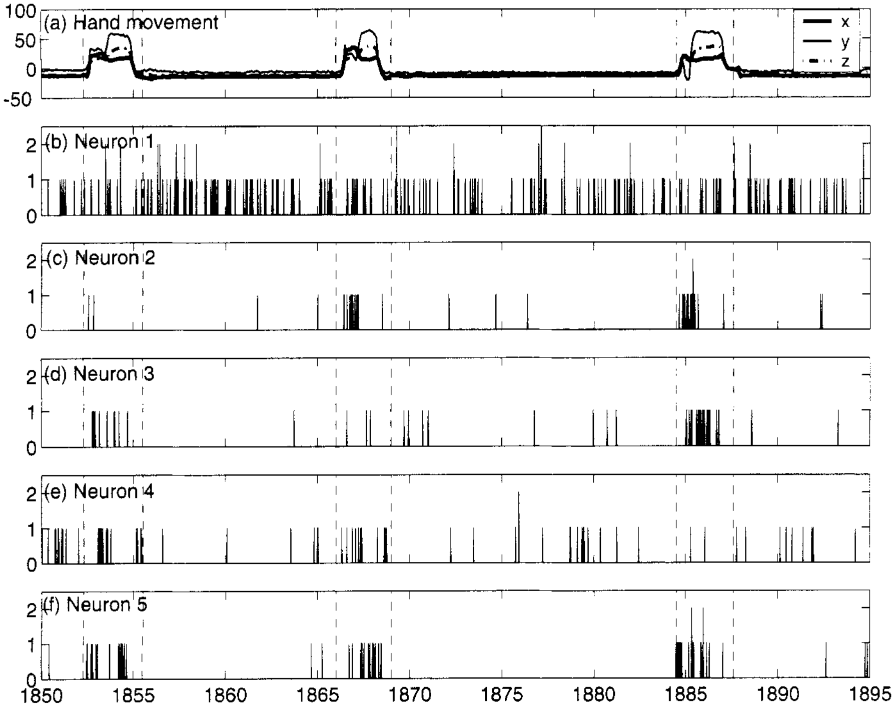

Figure 8.16. (a): X, Y, Z components of the monkey's hand movements. Dashed lines indicate time intervals when the monkey stretched its hand to grab food and subsequently placed the food in its mouth. (b–f): firings of five neurons associated with the hand movements plotted in (a).

8.9.2.2 Heterogeneity of neuronal firings revealed by distributional analysis Conventionally, neuronal interspike interval data are modeled by exponential and gamma distributions (see Eqs. (3.12) and (3.15)). Besides these two distributions, many other distributions have been observed from the monkey's neuronal firing data, such as log-normal (Eq. (3.20)) and power-law distributions (Eq. (3.23)). Four examples are shown in Figs. 8.17(a–d) for exponential, gamma, log-normal, and power-law distributions, respectively, for four different neurons. This simple distributional analysis clearly indicates that the interspike interval data of different neurons may follow very different distributions and, therefore, that the firing patterns of the neurons can be considered very heterogeneous. Existence of multiple distributions implies existence of different stochastic processes underlying neuronal firings in the cortex.

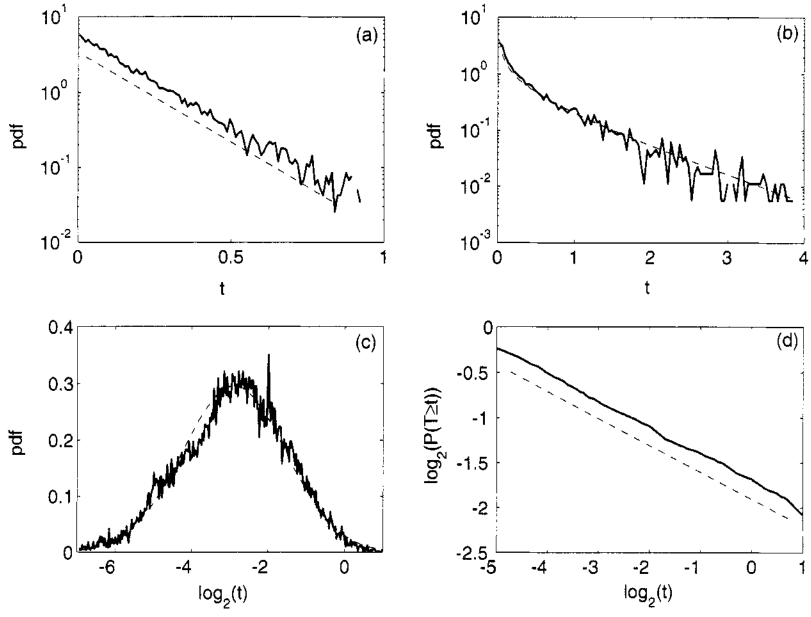

Figure 8.17. Four types of neuronal interspike interval distributions: (a) exponential, (b) gamma, (c) log-normal, and (d) power-law. Plotted in (a–c) and (d) are PDFs and the complementary cumulative distribution function (CCDF), respectively.

8.9.2.3 Nonstationary neuronal firings revealed by linear and nonlinear correlation analysis The nonstationarity of observed neuronal firing patterns can be quantified by characterizing the correlations between neuronal firings and hand trajectory. Simple linear correlations can be assessed by cross-correlation analysis between the firing of a specific neuron and the hand trajectory. More general correlations, including nonlinear correlations, can be characterized by mutual information. To compute the dependence of cross-correlation and mutual information with time, one can partition the data into many small segments, then calculate the correlations between the corresponding segments, and finally plot the correlation against the time index associated with each segment. Let the segment of firing data of a neuron be denoted by $w(t)$, and let the hand trajectory (either the x, y, or z component) be denoted by $u(t)$. The cross-correlation, denoted by $C(w, u)$, can be calculated by the simple equation

$$C(w,u) = max_L < w(t)u(t-L) >,$$

where $<>$ denotes the average within the segment and L is a small time (not necessarily positive) chosen in such a way that $< w(t)u(t-L) >$ is maximized. When $C(w,u)$ is normalized by the standard deviations of $w(t)$ and $u(t)$, one obtains the correlation coefficient. Before taking the average within each segment, one could remove the mean values of $w(t)$ and $u(t)$ first. The mutual information

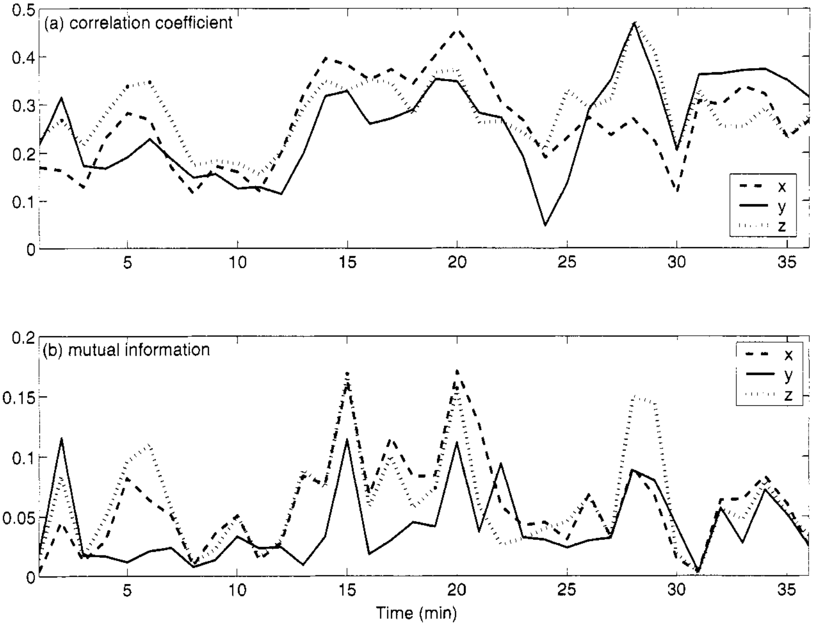

Figure 8.18. Time-varying correlations between spike counting data and hand movement data. (a) correlation coefficient; (b) mutual information.

of $w(t)$ and $u(t)$, written as $I(W, U)$, is the amount of information gained about u when w is learned, and vice versa. Let the probability distribution for $w(t)$ and $u(t)$ be denoted by $P(W = w_i)$, $i = 1, \cdots, N_w$, and $P(U = u_i)$, $i = 1, \cdots, N_u$, respectively. Then

$$\begin{aligned} I(W, U) &= H(U) - H(W|U) = H(W) - H(U|W) \\ &= H(W) + H(U) - H(W, U) \\ &= \sum_{i=1}^{N_w} \sum_{j=1}^{N_u} P(W = w_i, U = u_j) \ln \frac{P(W = w_i, U = u_j)}{P(W = w_i)P(U = u_j)}, \end{aligned} \tag{8.28}$$

where $H()$ denotes entropy. $I(W, U) = 0$ if and only if W and U are independent. The variations of correlation coefficient and mutual information with time are shown in Figs. 8.18(a) and 8.18(b), respectively. Evidently, both types of correlations vary considerably with time. Since hand trajectory is stationary, this indicates that neuronal firing patterns are highly nonstationary.

8.9.2.4 Long-range correlations in neuronal firings To gain further insights into the features defining the set of neurons that have strong correlations with hand trajectory (i.e., neurons similar to those shown in Figs. 8.16(c–f)), DFA is used to analyze spike-counting processes. The counting processes are obtained with the time window size $\Delta t = 0.1$ s. A few typical results are plotted in Fig. 8.19 for six neurons. It is observed that the lines are quite straight; therefore, the power-law relation of Eq. (8.22) is well defined. More interestingly, it is observed that three neurons, shown in Figs. 8.19(a–c), are characterized by a single fractal scaling, with the Hurst parameter ranging from about 0.5 to about 0.72, while three other neurons, shown in Figs. 8.19(d–f), have fractal scaling breaks around $m = 2^6 \sim 2^7$. Note that $m = 2^6 \sim 2^7$ corresponds to about $6.4 \sim 12.8$ s. For the range of m from 2^2 to about 2^7, the neurons shown in Figs. 8.19(d–f) have Hurst parameters all larger than 0.8. Interestingly, the time scale of $6.4 \sim 12.8$ s is comparable to the average time of 8 s between two successive reaching tasks (see Sec. A.3 of Appendix A). The neurons shown in Figs. 8.19(a–c) do not have much correlation with hand trajectory, just like the one shown in Fig. 8.16(b), while the neurons shown in Figs. 8.19(d–f) have very strong correlations with hand trajectory, like those plotted in Figs. 8.16(c–f). The large Hurst parameter and the time scale of $6.4 \sim 12.8$ s, therefore, strongly suggest that neurons well correlated with hand trajectory experienced a "resetting" effect at the start of each reaching task.

8.9.2.5 Dynamic coalition of neurons The above analyses clearly indicate that, in executing a movement task, different neurons have different degrees of importance and that each neuron's importance can also change with time. In other words, the execution of a movement task is carried out by a subset of "important" neurons in a specific cortical area but not by all the neurons. These neurons may be termed a *dynamic coalition of neurons*. A coalition of neurons means that many types of excitatory and inhibitory interconnected neurons are involved, which support one another, directly or indirectly by increasing the activity of their fellow members. "Dynamic" means that neurons within the coalition may leave the coalition and not participate in neural information processing, such as controlling hand movements. After they leave the coalition, they may rejoin it later or new neurons can join. The process of leaving and joining the coalition makes the structure of the coalition network vary greatly over time.

Intuitively, the concept of a coalition of neurons is very attractive, since it generalizes the concept of synchronization, a treasured one in neuroscience. The picture becomes overwhelming if one analyzes the neuronal firing patterns of a large number of neurons simultaneously measured, such as those shown in Fig. 8.16: In the time interval of three hand movements shown in Fig. 8.16, *if we define the dynamic coalition of neurons by their correlation with the hand trajectory, then neuron 1 does not belong to the coalition, neuron 5 belongs to the coalition, and neurons 2 to 4 are transient members — they do not belong to the coalition at the first, second, and third hand movement, respectively.* We emphasize that few neurons were active in all the hand movements. This resulted in highly nonstationary neuronal firing patterns.

It is interesting to note that a similar concept, termed dynamic core, has been put forward by the Nobel laureate Edelman and his colleague Tononi. The concept

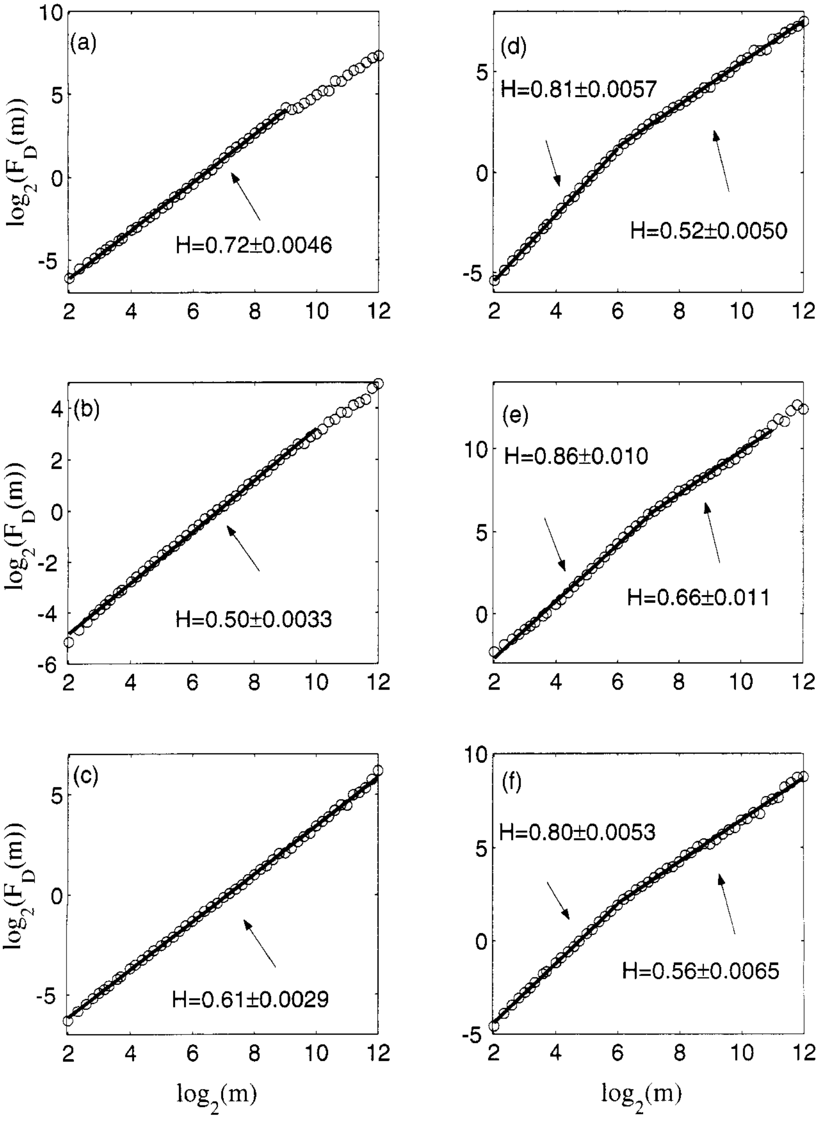

Figure 8.19. DFA for the spike count data: (a–c) neurons with one scaling range and (d–f) neurons with two scaling ranges.

of dynamic core has recently been rephrased as coalition of neurons by Crick and Koch. Recently, this concept has been enriched by a study of Gao et al. [151] on long-range correlations in multistable (or ambiguous) visual perception (see Fig. 8.20). The basic picture in that study is the following. During multistable visual perception, a sustained winning coalition of neurons may be responsible for one of the possible percepts formed. The winning coalition, however, may break down and subsequently be either replaced by a competing coalition or itself transformed into a different coalition. This corresponds to the switching of percepts. The memory may come from the inertia of the coalitions: a strong and stable coalition has to be won over by another similarly stable and strong coalition, resulting in long switching times. The stochasticity comes from the observation that each coalition is composed of a dynamic group of neurons. Being dynamic, the structure of the network of the coalitions of neurons, especially the connectivity of the network, must be highly transient. Such a coalition may involve a few or many groups of neurons in one or a few regions of the brain, while each group in a specific region is only a fraction of all of the neurons in that region.

The above arguments suggest that the simplest model of neural information processing has to contain two levels. One level defines a subset of significant neurons, whose firings are well correlated with movements. The other level defines a signal component through the collective behavior of the neurons belonging to the dynamic coalition of neurons. This is a largely uncharted territory. For relevant references, see Sec. 8.10.

8.9.3 Protein coding region identification

We now consider gene finding, one of the most important problems in the study of genomes. Current computational approaches for finding genes are based either on comparative search or Markov (or hidden Markov) models. Both approaches require considerable knowledge of the genome sequence under investigation. In order to be successful, a gene-finding algorithm has to incorporate good indices that best discriminate coding and noncoding regions. While a number of good codon indices have been proposed, most of them are in one way or another related to the period-3 (P3) feature of coding sequences. The P3 feature is due to the fact that three nucleotide bases encode an amino acid and that the usage of codons is highly biased. This feature can be manifested in a number of similar ways. For example, if one maps a DNA sequence of length N to a numerical sequence and then takes the Fourier transform, typically one observes a strong peak at (or around, if N is not a multiple of 3) $N/3$ in the magnitude of the Fourier transform if the sequence is a coding one. However, such a peak is either very weak or missing if the sequence is noncoding.

P3 characterizes certain regularity of a DNA sequence. A DNA sequence is, however, also random, with entropy per nucleotide base of around 1.9 bit. How can one simultaneously characterize both the P3 feature and the randomness of a DNA

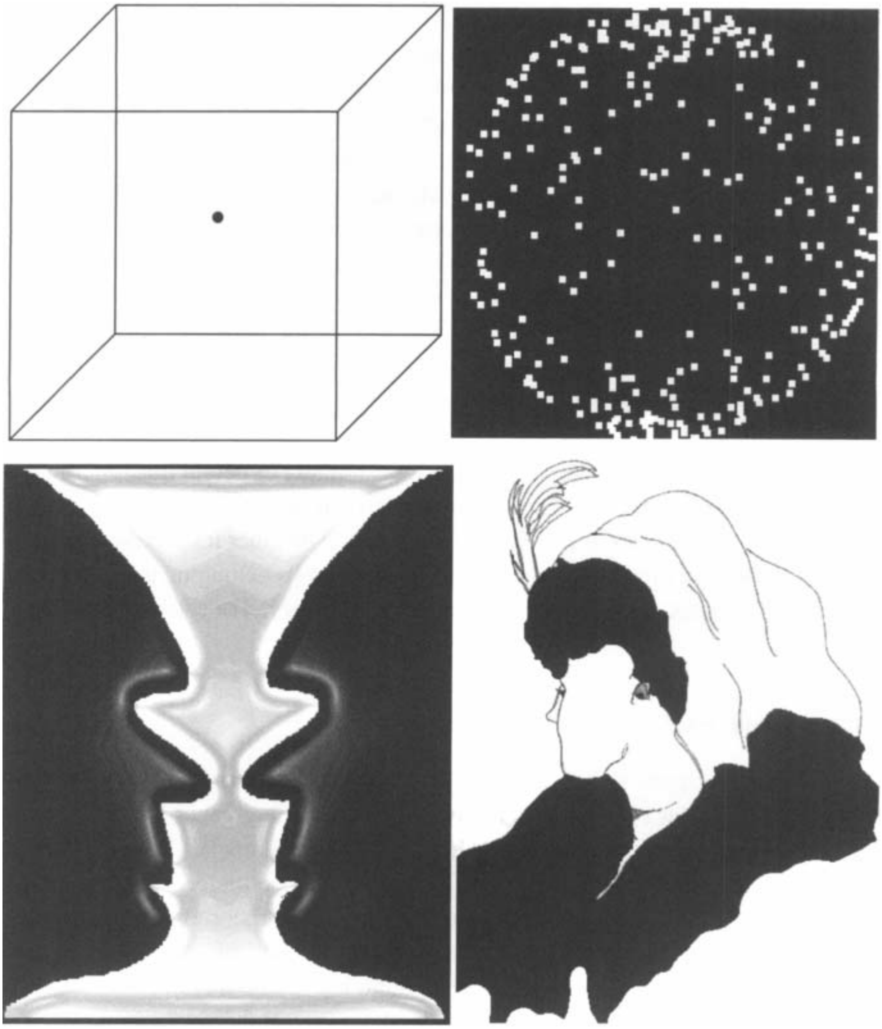

Figure 8.20. Well-known ambiguous figures. Upper left: Necker cube. When viewed steadily, the lower left side sometimes appears in the front of the cube, while at other times it appears to be in back. Upper right: turning ball. It may appear to turn horizontally or vertically. It is an example of binocular rivalry, a form of interocular competition that arises when the patterns in the two eyes cannot be fused stereoscopically. Lower left: Rubin's illustration of a vase and faces. Lower right: Boring's classic picture of a young girl and an old lady.

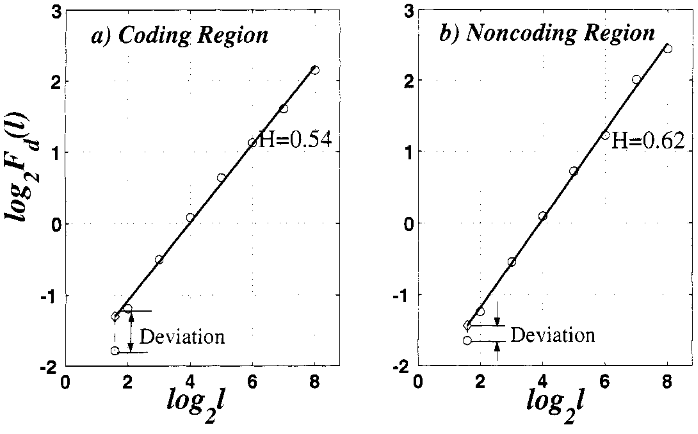

Figure 8.21. Representative P3 fractal deviation for coding (a) and noncoding (b) DNA.

sequence? One simple and effective method is to characterize the fractal scaling deviation at the specific scale of P3. To perform fractal analysis, one can first map a DNA sequence to a numerical sequence, then take partial summation to form a random walk, and finally apply multifractal DFA. As shown in Figs. 8.21(a,b), P3-induced deviation from fractal behavior is typically much larger for a coding sequence than for a noncoding sequence. When one employs the sliding window technique and partitions the P3 fractal-deviation curve into three subsets (i.e., $1, 4, 7, \cdots$; $2, 5, 8, \cdots$; $3, 6, 9, \cdots$, which correspond to three reading frames), one obtains Fig. 8.22, from which it is observed that in coding regions, the three reading frame–specific deviation curves separate, while in noncoding regions they mix together. By quantifying the separation of the three reading frame–specific deviation curves, two very effective codon indices can be devised. The percentage accuracy of one of them (FD) is shown in Table 8.1, where the parameters N_i, $i = 1, 2$ denote the number of coding and noncoding sequences of all 16 yeast chromosomes with length equal to or larger than n_i, $i = 1, 2$, respectively. Compared with methods based on P3 only (obtained by Fourier transform, denoted by FM in the table) or fractal only (denoted by H in the table), simultaneous characterization of P3 and fractal behavior is far more accurate.

8.10 BIBLIOGRAPHIC NOTES

The basic concept for the study of LRD time series was formulated by the distinguished statistician Cox [87] in 1984. The concept was later introduced to the study of network traffic by Leland et al. [281] and Beran et al. [44] and made more rigorous by Tsybakov and Georganas [452]. An early good reference on the many estimators of the Hurst parameters (excluding the wavelet based ones) is [424]. A

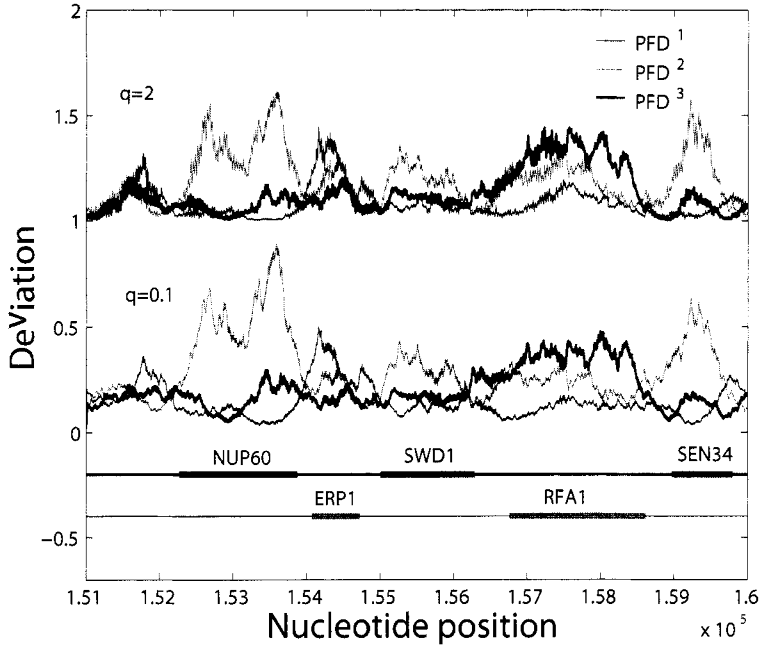

Figure 8.22. The reading frame–specific fractal deviation curves for a segment of DNA in yeast chromosome I (from nucleotide 151000 to 160000). Horizontal bars on the two lines below the deviation curves are the open reading frames on the two strands of the chromosome (first line: positive strand; second line: reverse strand).

more recent one is [157]. The wavelet-based estimator in traffic engineering was first proposed by Abry and Veitch [4]. The structure-function–based multifractal was initially developed for the study of turbulence [143]. DFA was originally proposed by Peng et al. [341]; see [422] for a discussion on the relation between DFA and PSD; for further developments of DFA, see [72, 73, 227, 245]. The l_1 norm formulation of multifractal DFA was developed in [162], while the l_2 norm formulation was developed in [246]. The AR(1) model as a pseudo-LRD traffic model was explored in [5]. It was (erroneously) claimed in [253] as an exact model for the $1/f$ process.

Readers interested in applications are referred to [235] on human heartbeat dynamics, to [388] on analysis of earthquake data, to [474] on human gait data, to [422, 454, 487] on analysis of geophysical data, to [214, 228] on sea clutter, to [70, 186, 259, 382, 383, 393, 396, 428, 471] on BMI, to [88, 117, 496] on dynamic coalition of neurons, and to [154, 159, 160] on protein-coding sequence identification (In order not to overwhelm readers with too many references, only a few are cited here. Readers may also refer to references cited by those works as well as those that have cited the references included here.)

Coding / Noncoding	Sensitivity/Specificity		
	FD	FM	H
(n_1, N_1) / (n_2, N_2)	w=64	w=63	w=64
(1, 4125) / (1, 5993)	82.6%	70.9%	43.9%
(256, 4067) / (256, 4164)	84.3%	71.5%	45.2%
(512, 3756) / (512, 1939)	87.3%	71.2%	45.5%
(1026, 2674) / (512, 1939)	89.2%	72.0%	45.1%
(1026, 2674) / (1026, 638)	92.4%	71.1%	43.7%

Table 8.1 Accuracy of the gene-finding algorithm

Readers interested in PCA of fractal processes are referred to [153]. For spectral analysis of networks, see [79, 110, 125, 126, 191, 193, 309]. Finally, for information on the four microarray datasets, see [9, 78, 196, 374].

8.11 EXERCISES

1. Write simple computer programs for the variance-time method, R/S statistic, FA, and DFA; apply them to the fGn/fBm data generated by you. If you are unable to generate the data, you may download some of them at http://www.gao.ece.ufl.edu/book_data/. Check out the methods by integrating/differencing the data.

2. Implement FA- and DFA-based multifractal analysis; apply your programs to your own data or those downloaded from the book's website.

3. Compute PSD for the AR(1) model by taking the Fourier transform of its autocorrelation function. Numerically generate a time series from the AR(1) model and compute its PSD. Check to determine if your simulation is consistent with Eq. (8.25).

4. Repeat the results plotted in Figs.8.5, 8.6, and 8.8 (hint: choose a suitable sampling time; do not make it either too small or too large).

5. Explain why the H from wavelet analysis of the u_i process (Fig. 8.15(b)) should be compared to the H from DFA of the v_i process (Fig. 8.14).