Chapter 1

Introduction

1.1 Biological Data in Digital Symbol Sequences

A fundamental feature of chain molecules, which are responsible for the function and evolution of living organisms, is that they can be cast in the form of digital symbol sequences. The nucleotide and amino acid monomers in DNA, RNA, and proteins are distinct, and although they are often chemically modified in physiological environments, the chain constituents can without infringement be represented by a set of symbols from a short alphabet. Therefore experimentally determined biological sequences can in principle be obtained with complete certainty. At a particular position in a *given* copy of a sequence we will find a distinct monomer, or letter, and not a mixture of several possibilities.

The digital nature of genetic data makes them quite different from many other types of scientific data, where the fundamental laws of physics or the sophistication of experimental techniques set lower limits for the uncertainty. In contrast, provided the economic and other resources are present, nucleotide sequences in genomic DNA, and the associated amino acid sequences in proteins, can be revealed completely. However, in genome projects carrying out large-scale DNA sequencing or in direct protein sequencing, a balance among purpose, relevance, location, ethics, and economy will set the standard for the quality of the data.

The digital nature of biological sequence data has a profound impact on the types of algorithms that have been developed and applied for computational analysis. While the goal often is to study a particular sequence and its molecular structure and function, the analysis typically proceeds through the study of an ensemble of sequences consisting of its different versions in different species, or even, in the case of polymorphisms, different versions in

the same species. Competent comparison of sequence patterns across species must take into account that biological sequences are inherently "noisy," the variability resulting in part from random events amplified by evolution. Because DNA or amino acid sequences with a given function or structure will differ (and be uncertain), sequence models *must be probabilistic*.

1.1.1 Database Annotation Quality

It is somehow illogical that although sequence data can be determined experimentally with high precision, they are generally not available to researchers without additional noise stemming from the joint effects of incorrect interpretation of experiments and incorrect handling and storage in public databases. Given that biological sequences are stored electronically, that the public databases are curated by a highly diverse group of people, and, moreover, that the data are annotated and submitted by an even more diverse group of biologists and bioinformaticians, it is perhaps understandable that in many cases the error rate arising from the subsequent handling of information may be much larger than the initial experimental error [100, 101, 327].

An important factor contributing to this situation is the way in which data are stored in the large sequence databases. Features in biological sequences are normally indicated by listing the relevant positions in numeric form, and not by the "content" of the sequence. In the human brain, which is renowned for its ability to handle vast amounts of information accumulated over the lifetime of the individual, information is recalled by content-addressable schemes by which a small part of a memory item can be used to retrieve its complete content. A song, for example, can often be recalled by its first two lines.

Present-day computers are designed to handle numbers—in many countries human "accession" numbers, in the form of Social Security numbers, for one thing, did not exist before them [103]. Computers do not like content-addressable procedures for annotating and retrieving information. In computer search passport attributes of people—their names, professions, and hair color—cannot always be used to single out a perfect match, and if at all most often only when formulated using correct language and perfect spelling.

Biological sequence retrieval algorithms can been seen as attempts to construct associative approaches for finding specific sequences according to an often "fuzzy" representation of their content. This is very different from the retrieval of sequences according to their functionality. When the experimentalist submits functionally relevant information, this information is typically converted from what in the laboratory is kept as marks, coloring, or scribbles on the sequence itself. This "semiotic" representation by content is then converted into a representation where integers indicate individual positions. The

numeric representation is subsequently impossible to review by human visual inspection.

In sequence databases, the result is that numerical feature table errors, instead of being acceptable noise on the retrieval key, normally will produce garbage in the form of more or less random mappings between sequence positions and the annotated structural or functional features. Commonly encountered errors are wrong or meaningless annotation of coding and noncoding regions in genomic DNA and, in the case of amino acid sequences, randomly displaced functional sites and posttranslational modifications. It may not be easy to invent the perfect annotation and data storage principle for this purpose. In the present situation it is important that the bioinformatician carefully take into account these potential sources of error when creating machine-learning approaches for prediction and classification.

In many sequence-driven mechanisms, certain nucleotides or amino acids are compulsory. Prior knowledge of this kind is an easy and very useful way of catching typographical errors in the data. It is interesting that machine-learning techniques provide an alternative and also very powerful way of detecting erroneous information and annotation. In a body of data, if something is notoriously hard to learn, it is likely that it represents either a highly atypical case or simply a wrong assignment. In both cases, it is nice to be able to sift out examples that deviate from the general picture. Machine-learning techniques have been used in this way to detect wrong intron splice sites in eukaryotic genes [100, 97, 101, 98, 327], wrong or missing assignments of O-linked glycosylation sites in mammalian proteins [235], or wrongly assigned cleavage sites in polyproteins from picornaviruses [75], to mention a few cases. Importantly, not all of the errors stem from data handling, such as incorrect transfer of information from published papers into database entries: significant number of errors stems from incorrect assignments made by experimentalists [327]. Many of these errors could also be detected by simple consistency checks prior to incorporation in a public database.

A general problem in the annotation of the public databases is the fuzzy statements in the entries regarding *who* originally produced the feature annotation they contain. The evidence may be experimental, or assigned on the basis of sequence similarity or by a prediction algorithm. Often ambiguities are indicated in a hard-to-parse manner in free text, using question marks or comments such as POTENTIAL or PROBABLE. In order not to produce *circular* evaluation of the prediction performance of particular algorithms, it is necessary to prepare the data carefully and to discard data from unclear sources. Without proper treatment, this problem is likely to increase in the future, because more prediction schemes will be available. One of the reasons for the success of machine-learning techniques within this imperfect data domain is that the methods often—in analogy to their biological counterparts—are able

to handle noise, provided large corpora of sequences are available. New discoveries within the related area of natural language acquisition have proven that even eight-month-old infants can detect linguistic regularities and *learn* simple statistics for the recognition of word boundaries in continuous speech [458]. Since the language the infant has to learn is as unknown and complex as the DNA sequences seem to us, it is perhaps not surprising that learning techniques can be useful for revealing similar regularities in genomic data.

1.1.2 Database Redundancy

Another recurrent problem haunting the analysis of protein and DNA sequences is the redundancy of the data. Many entries in protein or genomic databases represent members of protein and gene families, or versions of homologous genes found in different organisms. Several groups may have submitted the same sequence, and entries can therefore be more or less closely related, if not identical. In the best case, the annotation of these very similar sequences will indeed be close to identical, but significant differences may reflect genuine organism or tissue specific variation.

In sequencing projects redundancy is typically generated by the different experimental approaches themselves. A particular piece of DNA may for example be sequenced in genomic form as well as in the form of cDNA complementary to the transcribed RNA present in the cell. As the sequence being deposited in the databases is determined by widely different approaches—ranging from noisy single-pass sequence to finished sequence based on five- to tenfold repetition—the same gene may be represented by many database entries displaying some degree of variation.

In a large number of eukaryotes, the cDNA sequences (complete or incomplete) represent the spliced form of the pre-mRNA, and this means again, for genes undergoing *alternative splicing*, that a given piece of genomic DNA in general will be associated with several cDNA sequences being noncontinuous with the chromosomal sequence [501]. Alternative splice forms can be generated in many different ways. Figure 1.1 illustrates some of the different ways coding and noncoding segments may be joined, skipped, and replaced during splicing. Organisms having a splice machinery at their disposal seem to use alternative splicing quite differently. The alternative to alternative splicing is obviously to include different versions of the same gene as individual genes in the genome. This may be the strategy used by the nematode *Caenorhabditis elegans*, which seems to contain a large number of genes that are very similar, again giving rise to redundancy when converted into data sets [315]. In the case of the human genome [234, 516, 142] it is not unlikely that at least 30-80% of the genes are alternatively spliced, in fact it may be the rule rather than

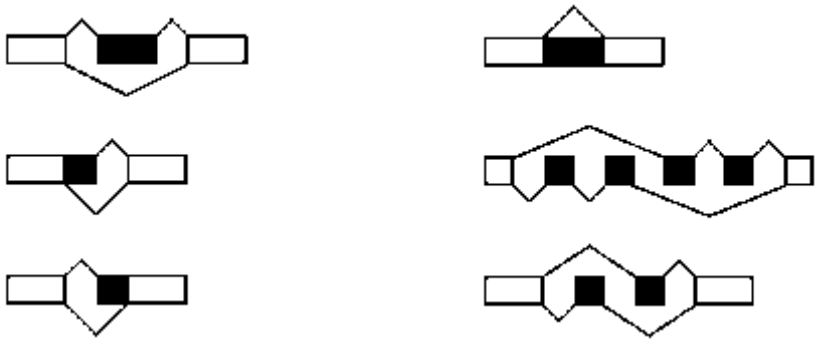

Figure 1.1: The Most Common Modes of Alternative Splicing in Eukaryotes. Left from top: Cassette exon (exon skipping or inclusion), alternative 5' splice site, alternative 3' splice site. Right from top: whole intron retention, pairwise spliced exons and mutually exclusive exons. These different types of alternative pre-mRNA processing can be combined [332].

the exception.

Data redundancy may also play a nontrivial role in relation to massively parallel gene expression experiments, a topic we return to in chapter 12. The sequence of genes either being spotted onto glass plates, or synthesized on DNA chips, is typically based on sequences, or clusters of sequences, deposited in the databases. In this way microarrays or chips may end up containing more sequences than there are genes in the genome of a particular organism, thus giving rise to noise in the quantitative levels of hybridization recorded from the experiments.

In protein databases a given gene may also be represented by amino acid sequences that do not correspond to a direct translation of the genomic wild-type sequence of nucleotides. It is not uncommon that protein sequences are modified slightly in order to obtain sequence versions that for example form better crystals for use in protein structure determination by X-ray crystallography [99]. Deletions and amino acid substitutions may give rise to sequences that generate database redundancy in a nontrivial manner.

The use of a redundant data set implies at least three potential sources of error. First, if a data set of amino acid or nucleic acid sequences contains large families of closely related sequences, statistical analysis will be biased toward these families and will overrepresent features peculiar to them. Second, apparent correlations between different positions in the sequences may be an artifact of biased sampling of the data. Finally, if the data set is being used for predicting a certain feature and the sequences used for making and calibrating the prediction method—the training set—are too closely related to

the sequences used for testing, the apparent predictive performance may be overestimated, reflecting the method's ability to reproduce its own particular input rather than its generalization power.

At least some machine-learning approaches will run into trouble when certain sequences are heavily overrepresented in a training set. While algorithmic solutions to this problem have been proposed, it may often be better to clean up the data set first and thereby give the underrepresented sequences equal opportunity. It is important to realize that underrepresentation can pose problems both at the primary structure level (sequence redundancy) and at the classification level. Categories of protein secondary structures, for example, are typically skewed, with random coil being much more frequent than beta-sheet.

For these reasons, it can be necessary to avoid too closely related sequences in a data set. On the other hand, a too rigorous definition of "too closely related" may lead to valuable information being discarded from the data set. Thus, there is a trade-off between data set size and nonredundancy. The appropriate definition of "too closely related" may depend strongly on the problem under consideration. In practice, this is rarely considered. Often the test data are described as being selected "randomly" from the complete data set, implying that great care was taken when preparing the data, even though redundancy reduction was not applied at all. In many cases where redundancy reduction is applied, either a more or less arbitrary similarity threshold is used, or a "representative" data set is made, using a conventional list of protein or gene families and selecting one member from each family.

An alternative strategy is to keep all sequences in a data set and then assign weights to them according to their novelty. A prediction on a closely related sequence will then count very little, while the more distantly related sequences may account for the main part of the evaluation of the predictive performance. A major risk in this approach is that erroneous data almost always will be associated with large weights. Sequences with erroneous annotation will typically stand out, at least if they stem from typographical errors in the feature tables of the databases. The prediction for the wrongly assigned features will then have a major influence on the evaluation, and may even lead to a drastic underestimation of the performance. Not only will false sites be very hard to predict, but the true sites that would appear in a correct annotation will often be counted as false positives.

A very productive way of exploiting database redundancy—both in relation to sequence retrieval by alignment and when designing input representations for machine learning algorithms—is the *sequence profile* [226]. A profile describes position by position the amino acid variation in a family of sequences organized into a multiple alignment. While the profile no longer contains information about the sequential pattern in individual sequences, the degree of sequence variation is extremely powerful in database search, in programs such

as PSI-BLAST, where the profile is iteratively updated by the sequences picked up by the current version of the profile [12]. In later chapters, we shall return to hidden Markov models, which also implement the profile concept in a very flexible manner, as well as neural networks receiving profile information as input—all different ways of taking advantage of the redundancy in the information being deposited in the public databases.

1.2 Genomes—Diversity, Size, and Structure

Genomes of living organisms have a profound diversity. The diversity relates not only to genome size but also to the storage principle as either single- or double-stranded DNA or RNA. Moreover, some genomes are linear (e.g. mammals), whereas others are closed and circular (e.g. most bacteria).

Cellular genomes are always made of DNA [389], while phage and viral genomes may consist of either DNA or RNA. In single-stranded genomes, the information is read in the positive sense, the negative sense, or in both directions, in which case one speaks of an ambisense genome. The positive direction is defined as going from the 5' to the 3' end of the molecule. In double-stranded genomes the information is read only in the positive direction (5' to 3' on either strand). Genomes are not always replicated directly; retroviruses, for example, have RNA genomes but use a DNA intermediate in the replication.

The smallest genomes are found in nonself-replicating suborganisms like bacteriophages and viruses, which sponge on the metabolism and replication machinery of free-living prokaryotic and eukaryotic cells, respectively. In 1977, the 5,386 bp in the genome of the bacteriophage ϕX174 was the first to be sequenced [463]. Such very small genomes normally come in one continuous piece of sequence. But other quite small genomes, like the 1.74 Mbp genome of the hyperthermophilic archaeon *Methanococcus jannaschii*, which was completely sequenced in 1996, may have several chromosomal components. In *M. jannaschii* there are three, one of them by far the largest. The much larger 3,310 Mbp human genome is organized into 22 chromosomes plus the two that determine sex. Even among the primates there is variation in the number of chromosomes. Chimpanzees, for example, have 23 chromosomes in addition to the two sex chromosomes. The chimpanzee somatic cell nucleus therefore contains a total number of 48 chromosomes in contrast to the 46 chromosomes in man. Other mammals have completely different chromosome numbers, the cat, for example, has 38, while the dog has as many as 78 chromosomes. As most higher organisms have two near-identical copies of their DNA (the *diploid* genome), one also speaks about the *haploid* DNA content, where only one of the two copies is included.

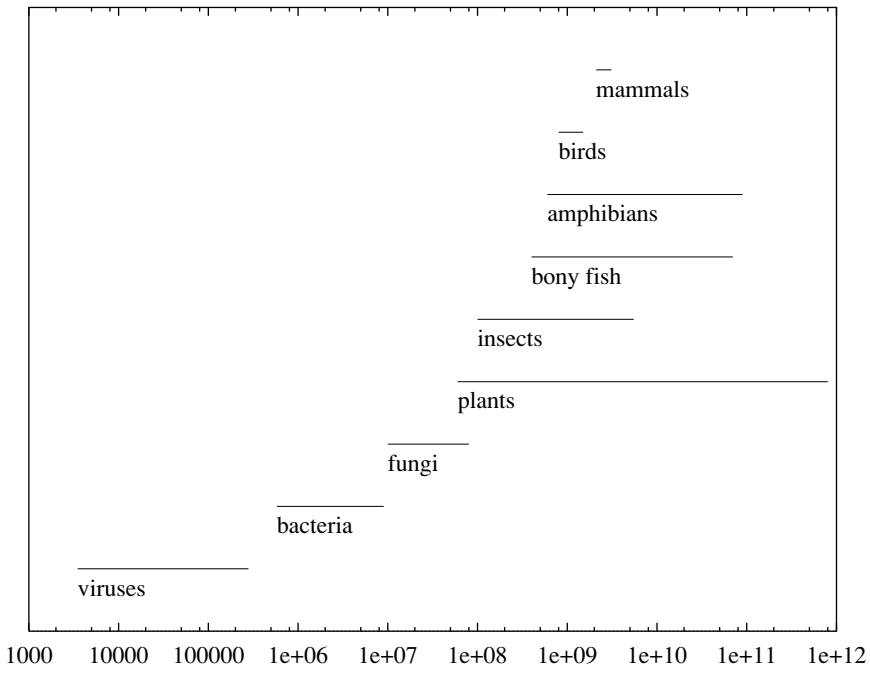

Figure 1.2: Intervals of Genome Sizes for Various Classes of Organisms. Note that the plot is logarithmic in the number of nucleotides on the first axis. Most commonly, the variation within one group is one order of magnitude or more. The narrow interval of genome sizes among mammals is an exception to the general picture. It is tempting to view the second axis as "organism complexity" but it is most certainly not a direct indication of the size of the gene pool. Many organisms in the upper part of the spectrum, e.g., mammals, fish, and plants, have comparable numbers of genes (see table 1.1).

The chromosome in some organisms is not stable. For example, the *Bacillus cereus* chromosome has been found to consist of a large stable component (2.4 Mbp) and a smaller (1.2 Mbp) less stable component that is more easily mobilized into extra-chromosomal elements of varying sizes up to the order of megabases [114]. This has been a major obstacle in determining the genomic sequence, or just a genetic map, of this organism. However, in almost any genome transposable elements can also be responsible for rearrangements, or insertion, of fairly large sequences, although they have been not been reported to cause changes in chromosome number. Some theories claim that a high number of chromosomal components is advantageous and increases the speed of evolution, but currently there is no final answer to this question [438].

It is interesting that the spectrum of genome sizes is to some extent segregated into nonoverlapping intervals. Figure 1.2 shows that viral genomes have sizes in the interval from 3.5 to 280 Kbp, bacteria range from 0.5 to 10 Mbp, fungi from around 10 to 50 Mbp, plants start at around 50 Mbp, and mammals are found in a more narrow band (on the logarithmic scale) around 1 Gb. This staircase reflects the sizes of the gene pools that are necessary for maintaining life in a noncellular form (viruses), a unicellular form (bacteria), multicellular forms without sophisticated intercellular communication (fungi), and highly differentiated multicellular forms with many intercellular signaling systems (mammals and plants). In recent years it has been shown that even bacteria are capable of chemical communication [300]. Molecular messengers may travel between cells and provide populationwide control. One famous example is the expression of the enzyme luciferase, which along with other proteins is involved in light production by marine bacteria. Still, this type of communication requires a very limited gene pool compared with signaling in higher organisms.

The general rule is that within most classes of organisms we see a huge relative variation in genome size. In eukaryotes, a few exceptional classes (e.g., mammals, birds, and reptiles) have genome sizes confined to a narrow interval [116]. As it is possible to estimate the size of the unsequenced gaps, for example by optical mapping, the size of the human genome is now known with a quite high precision. Table 1.2 shows an estimate of the size for each of the 24 chromosomes. In total the reference human genome sequence seems to contain roughly 3,310,004,815 base pairs—an estimate that presumably will change slightly over time.

The cellular DNA content of different species varies by over a millionfold. While the size of bacterial genomes presumably is directly related to the level of genetic and organismic complexity, within the eukaryotes there might be as much as a 50,000-fold excess compared with the basic protein-coding requirements [116]. Organisms that basically need the same molecular apparatus can have a large variation in their genome sizes. Vertebrates share a lot of basic machinery, yet they have very different genome sizes. As early as 1968, it was demonstrated that some fish, in particular the family Tetraodontidae, which contains the pufferfish, have very small genomes [254, 92, 163, 534, 526]. The pufferfish have genomes with a haploid DNA content around 400–500 Mbp, six–eight times smaller than the 3,310 Mbp human genome. The pufferfish *Fugu rubripes* genome is only four times larger than that of the much simpler nematode worm *Caenorhabditis elegans* (100 Mbp) and eight times smaller than the human genome. The vertebrates with the largest amount of DNA per cell are the amphibians. Their genomes cover an enormous range, from 700 Mbp to more than 80,000 Mbp. Nevertheless, they are surely less complex than most humans in their structure and behavior [365].

Group	Species	Genes	Genome size
Phages	Bacteriophage MS2	4	0.003569
	Bacteriophage T4	270	0.168899
Viruses	Cauliflower mosaic virus	8	0.008016
	HIV type 2	9	0.009671
	Vaccinia virus	260	0.191737
Bacteria	*Mycoplasma genitalium*	473	0.58
	Mycoplasma pneumoniae	716	0.82
	Haemophilus influenzae	1,760	1.83
	Bacillus subtilis	3,700	4.2
	Escherichia coli	4,100	4.7
	Myxococcus xanthus	8,000	9.45
Archaea	*Methanococcus jannaschii*	1,735	1.74
Fungi	*Saccharomyces cerevisiae*	5,800	12.1
Protoctista	*Cyanidioschyzon merolae*	5,000	11.7
	Oxytricha similis	12,000	600
Arthropoda	*Drosophila melanogaster*	15,000	180
Nematoda	*Caenorhabditis elegans*	19,000	100
Mollusca	*Loligo pealii*	20-30,000	2,700
Plantae	*Nicotiana tabacum*	20-30,000	4,500
	Arabidopsis thaliana	25,500	125
Chordata	*Giona intestinalis*	N	165
	Fugu rubripes	30-40,000	400
	Danio rerio	N	1,900
	Mus musculus	30-40,000	3,300
	Homo sapiens	30-40,000	3,310

Table 1.1: Approximate Gene Number and Genome Sizes in Organisms in Different Evolutionary Lineages. Genome sizes are given in megabases. N = not available. Data were taken in part from [390] and references therein (and scaled based on more current estimates); others were compiled from a number of different Internet resources, papers, and books.

1.2.1 Gene Content in the Human Genome and other Genomes

A variable part of the complete genome sequence in an organism contains *genes*, a term normally defined as one or several segments that constitute an expressible unit. The word *gene* was coined in 1909 by the Danish geneticist Wilhelm Johannsen (together with the words genetype and phenotype) long before the physical basis of DNA was understood in any detail.

Genes may encode a protein product, or they may encode one of the many RNA molecules that are necessary for the processing of genetic material and for the proper functioning of the cell. mRNA sequences in the cytoplasm are used as recipes for producing many copies of the same protein; genes encoding other RNA molecules must be transcribed in the quantities needed. Se-

Human chromosome	Size
Chr. 1	282,193,664
Chr. 2	253,256,583
Chr. 3	227,524,578
Chr. 4	202,328,347
Chr. 5	203,085,532
Chr. 6	182,415,242
Chr. 7	166,623,906
Chr. 8	152,776,421
Chr. 9	142,271,444
Chr. 10	145,589,288
Chr. 11	150,783,553
Chr. 12	144,282,489
Chr. 13	119,744,898
Chr. 14	106,953,321
Chr. 15	101,380,521
Chr. 16	104,298,331
Chr. 17	89,504,553
Chr. 18	86,677,548
Chr. 19	74,962,845
Chr. 20	66,668,005
Chr. 21	44,907,570
Chr. 22	47,662,662
Chr. X	162,599,930
Chr. Y	51,513,584

Table 1.2: Approximate Sizes for the 24 Chromosomes in the Human Genome Reference Sequence. Note that the 22 chromosome sizes do not rank according to the original numbering of the chromosomes. Data were taken from the Ensembl (www.ensembl.org) and Santa Cruz (genome.ucsc.edu) web-sites. In total the reference human genome sequence seems to contain roughly 3,310,004,815 base pairs—an estimate that presumably will change slightly over time.

quence segments that do not directly give rise to gene products are normally called noncoding regions. Noncoding regions can be parts of genes, either as regulatory elements or as intervening sequences interrupting the DNA that directly encode proteins or RNA. Machine-learning techniques are ideal for the hard task of interpreting unannotated genomic DNA, and for distinguishing between sequences with different functionality.

Table 1.1 shows the current predictions for the approximate number of genes and the genome size in organisms in different evolutionary lineages. In those organisms where the complete genome sequence has now been determined, the indications of these numbers are of course quite precise, while in other organisms only a looser estimate of the gene density is available. In some

Species	Haploid genome size	Bases	Entries
Homo sapiens	3,310,000,000	7,387,490,518	4,544,962
Mus musculus	3,300,000,000	1,527,228,639	2,793,543
Drosophila melanogaster	180,000,000	502,655,942	167,687
Arabidopsis thaliana	125,000,000	249,689,164	183,987
Caenorhabditis elegans	100,000,000	204,396,881	114,744
Oryza sativa	400,000,000	171,870,798	161,411
Tetraodon nigroviridis	350,000,000	165,542,107	189,000
Rattus norvegicus	2,900,000,000	114,331,466	229,838
Bos taurus	3,600,000,000	76,700,774	168,469
Glycine max	1,115,000,000	73,450,470	167,090
Medicago truncatula	400,000,000	60,606,228	120,670
Lycopersicon esculentum	655,000,000	56,462,749	109,913
Trypanosoma brucei	35,000,000	50,723,464	91,360
Hordeum vulgare	5,000,000,000	49,770,458	70,317
Giardia intestinalis	12,000,000	49,431,105	56,451
Strongylocentrotus purpur	900,000,000	47,633,412	77,554
Danio rerio	1,900,000,000	47,584,911	93,141
Xenopus laevis	3,100,000,000	46,517,145	92,041
Zea mays	5,000,000,000	45,978,459	98,818
Entamoeba histolytica	20,000,000	44,552,032	49,969

Table 1.3: The Number of Bases in GenBank rel. 123, April 2001, for the 20 Most Sequenced Organisms. For some organisms there is far more sequence than the size of the genome, due to strain variation and pure redundancy.

organisms, such as bacteria, where the genome size is a strong growth-limiting factor, almost the entire genome is covered with coding (protein and RNA) regions; in other, more slowly growing organisms the coding part may be as little as 1–2%. This means that the gene density in itself normally will influence the precision with which computational approaches can perform gene finding. The noncoding part of a genome will often contain many pseudo-genes and other sequences that will show up as false positive predictions when scanned by an algorithm.

The biggest surprise resulting from the analysis of the two versions of the human genome data [134, 170] was that the gene content may be as low as in the order of 30,000 genes. Only about 30,000-40,000 genes were estimated from the initial analysis of the sequence. It was not totally unexpected as the gene number in the fruit fly (14,000) also was unexpectedly low [132]. But how can man realize its biological potential with less than twice the number of genes found in the primitive worm *C. elegans*? Part of the answer lies in alternative splicing of this limited number of genes as well as other modes of multiplexing the function of genes. This area has to some degree been ne-

glected in basic research and the publication of the human genome illustrated our ignorance all too clearly: only a year before the publication it was expected that around 100-120,000 genes would be present in the sequence [361]. For a complex organism, gene multiplexing makes it possible to produce several different transcripts from many of the genes in its genome, as well as many diferent protein variants from each transcript. As the cellular processing of genetic material is far more complex (in terms of regulation) than previously believed the need for sophisticated bioinformatics approaches with ability to model these processes is also strongly increased.

One of the big open questions is clearly how a quite substantial increase in organism complexity can arise from a quite modest increase in the size of the gene pool. The fact that worms have almost as many genes as humans is somewhat irritating, and in the era of whole cell and whole organism oriented research, we need to understand how the organism complexity scales with the potential of a fixed number of genes in a genome.

The French biologist Jean-Michel Claverie has made [132] an interesting "personal" estimate of the biological complexity K and its relation to the number of genes in a genome, N. The function f that converts N into K could in principle be linear ($K \sim N$), polynomial ($K \sim N^a$), exponential ($K \sim a^N$), $K \sim N!$ (factorial), and so on. Claverie suggests that the complexity should be related to the organism's ability to create diversity in its gene expression, that is to the number of theoretical transcriptome states the organism can achieve. In the simplest model, where genes are assumed to be either active or inactive (ON or OFF), a genome with N genes can potentially encode 2^N states. When we then compare humans to worms, we appear to be

$$2^{30,000}/2^{20,000} \cong 10^{3,000} \tag{1.1}$$

more complex than nematodes thus confirming (and perhaps reestablishing) our subjective view of superiority of the human species. In this simple model the exponents should clearly be decreased because genes are not independently expressed (due to redundance and/or coregulation), and the fact that many of the states will be lethal. On the other hand gene expression is not ON/OFF, but regulated in a much more graded manner. A quite trivial mathematical model can thus illustrate how a small increase in gene number can lead to a large increase in complexity and suggests a way to resolve the apparent N value paradox which has been created by the whole genome sequencing projects. This model based on patterns of gene expression may seem very trivial, still it represents an attempt to quantify "systemic" aspects of organisms, even if all their parts still may be understood using more conventional, reductionistic approaches [132].

Another fundamental and largely unsolved problem is to understand why the part of the genome that code for protein, in many higher organisms, is

quite limited. In the human sequence the coding percentage is small no matter whether one uses the more pessimistic gene number N of 26,000 or the more optimistic figure of 40,000 [170]. For these two estimates in the order of 1.1% (1.4%) of the human sequence seems to be coding, with introns covering 25% (36%) and the remaining intergenic part covering 75% (64%), respectively. While it is often stated that the genes only cover a few percent, this is obviously not true due to the large average intron size in humans. With the estimate of 40,000 genes more than one third of the entire human genome is covered by genes.

The mass of the nuclear DNA in an unreplicated haploid genome in a given organism is known as its C-value, because it usually is a constant in any one narrowly defined type of organism. The C-values of eukaryotic genomes vary at least 80,000-fold across species, yet bear little or no relation to organismic complexity or to the number of protein-coding genes [412, 545]. This phenomenon is known as the C-value paradox [518].

It has been suggested that noncoding DNA just accumulates in the nuclear genome until the costs of replicating it become too great, rather than having a structural role in the nucleus [412]. It became clear many years ago that the extra DNA does not in general contain an increased number of genes. If the large genomes contained just a proportionally increased number of copies of each gene, the kinetics of DNA renaturation experiments would be very fast. In renaturation experiments a sample of heat-denatured strands is cooled, and the strands reassociate provided they are sufficiently complementary. It has been shown that the kinetics is reasonably slow, which indicates that the extra DNA in voluminous genomes most likely does not encode genes [116]. In plants, where some of the most exorbitant genomes have been identified, clear evidence for a correlation between genome size and climate has been established [116]; the very large variation still needs to be accounted for in terms of molecular and evolutionary mechanisms. In any case, the size of the complete message in a genome is not a good indicator of the "quality" of the genome and its efficiency.

This situation may not be as unnatural as it seems. In fact, it is somewhat analogous to the case of communication between humans, where the message length fails to be a good measure of the quality of the information exchanged. Short communications can be very efficient, for example, in the scientific literature, as well as in correspondence between collaborators. In many E-mail exchanges the "garbage" has often been reduced significantly, leaving the essentials in a quite compact form. The shortest known correspondence between humans was extremely efficient: Just after publishing *Les Misérables* in 1862, Victor Hugo went on holiday, but was anxious to know how the sales were going. He wrote a letter to his publisher containing the single symbol "?". The publisher wrote back, using the single symbol "!", and Hugo could continue his

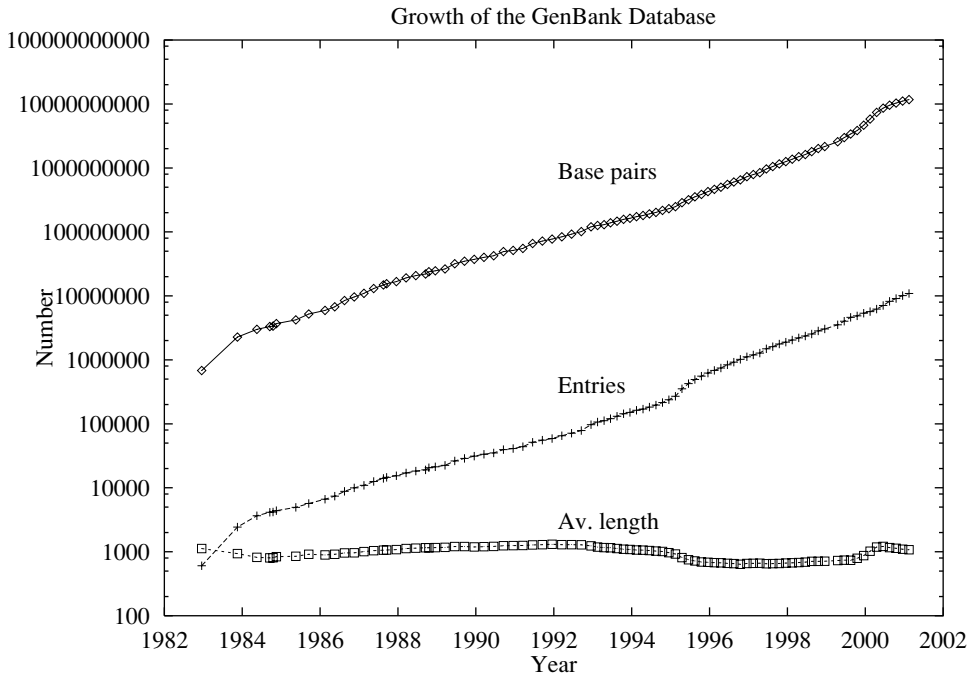

Figure 1.3: The Exponential Growth in the Size of the GenBank Database in the Period 1983-2001. Based on the development in 2000/2001, the doubling time is around 10 months. The complete size of GenBank rel. 123 is 12,418,544,023 nucleotides in 11,545,572 entries (average length 1076). Currently the database grows by more than 11,000,000 bases per day.

holiday without concern for this issue. The book became a best-seller, and is still a success as a movie and a musical.

The exponential growth in the size of the GenBank database [62, 503] is shown in figure 1.3. The 20 most sequenced organisms are listed in table 1.3. Since the data have been growing exponentially at the same pace for many years, the graph will be easy to extrapolate until new, faster, and even more economical sequencing techniques appear. If completely new sequencing approaches are invented the growth rate will presumably increase even further. Otherwise, it is likely that the rate will stagnate when several of the mammalian genomes have been completed. If sequencing at that time is still costly, funding agencies may start to allocate resources to other scientific areas, resulting in a lower production rate.

In addition to the publicly available data deposited in GenBank, proprietary data in companies and elsewhere are also growing at a very fast rate. This

means that the current total amount of sequence data known to man is unknown. Today the raw sequencing of a complete prokaryotic genome may—in the largest companies—take less than a day, when arrays of hundreds of sequencing machines are operating in parallel on different regions of the same chromosome. Part of this kind of data will eventually be deposited in the public databases, while the rest will remain in the private domain. For all organisms speed matters a lot, not the least due to the patenting that usually is associated with the generation of sequence data.

1.3 Proteins and Proteomes

1.3.1 From Genome to Proteome

At the protein level, large-scale analysis of complete genomes has its counterpart in what has become known as *proteome* analysis [299, 413]. Proteomes contain the total protein expression of a set of chromosomes. In a multicellular organism this set of proteins will differ from cell type to cell type, and will also change with time because gene regulation controls advances in development from the embryonic stage and further on. Proteome research deals with the proteins produced by genes from a given genome.

Unlike the word "genome" which was coined just after the First World War by the German botanist Hans Winkler [561, 65], the word "proteome" entered the scientific literature recently, in 1994 in papers by Marc Wilkins and Keith Williams [559].

Proteome analysis not only deals with determining the sequence, location, and function of protein-encoding genes, but also is strongly concerned with the precise biochemical state of each protein in its posttranslational form. These active and functional forms of proteins have in several cases been successfully predicted using machine-learning techniques.

Proteins often undergo a large number of modifications that alter their activities. For example, certain amino acids can be linked covalently (or noncovalently) to carbohydrates, and such amino acids represent so-called *glycosylation* sites. Other amino acids are subjected to *phosphorylation*, where phosphate groups are added to the polypeptide chain. In both cases these changes, which are performed by a class of specific enzymes, may be essential for the functional role of the protein. Many other types of posttranslational modifications exist, such as addition of fatty acids and the cleavage of signal peptides in the N-terminus of secretory proteins translocated across a membrane. Together with all the other types, these modifications are very interesting in a data-driven prediction context, because a relatively large body of experimentally verified sites and sequences is deposited in the public databases.

1.3.2 Protein Length Distributions

The evolution of living organisms selects polypeptide chains with the ability to acquire stable conformations in the aqueous or lipid environments where they perform their function. It is well known that interaction between residues situated far from each other in the linear sequence of amino acids plays a crucial role in the folding of proteins. These long-range effects also represent the major obstacle to computational approaches to protein folding. Still, most research on the topic concentrates on the local aspects of the structure elucidation problem. This holds true for strategies involving prediction and classification as well as for computational approaches based on molecular forces and the equations of motion.

Statistical analysis has played a major role in studies of protein sequences and their evolution since the early studies of Ycas and Gamow [195, 575, 555]. Most work has focused on the statistics of local nonrandom patterns with a specific structure or function, while reliable global statistics of entire genomes have been made possible by the vast amounts of data now available.

The universe of protein sequences can be analyzed in its entirety across species, but also in an organism-specific manner where, for example, the length distribution of the polypeptide chains in the largest possible proteome can be identified completely. A key question is whether the protein sequences we see today represent "edited" versions of sequences that were of essentially random composition when evolution started working on them [555]. Alternatively, they could have been created early on with a considerable bias in their composition.

Using the present composition of soluble proteins, one can form on the order of 10^{112} "natural" sequences of length-100 amino acids. Only a very tiny fraction of these potential sequences has been explored by Nature. A "random origin hypothesis," which asserts that proteins originated by stochastic processes according to simple rules, has been put forward by White and Jacobs [556, 555]. This theory can be taken formally as a null hypothesis when examining different aspects of the randomness of protein sequences, in particular to what extent proteins can be distinguished from random sequences.

The evidence for long-range order and regularity in protein primary structure is accumulating. Surprisingly, species-specific regularity exists even at a level below the compositional level: the typical length of prokaryotic proteins is consistently different from the typical length in eukaryotes [64]. This may be linked to the idea that the probability of folding into a compact structure increases more rapidly with length for eukaryotic than for prokarytic sequences [555]. It has been suggested that the observed differences in the sequence lengths can be explained by differences in the concentration of disulfide bonds between cysteine residues and its influence on the optimal domain sizes [304].

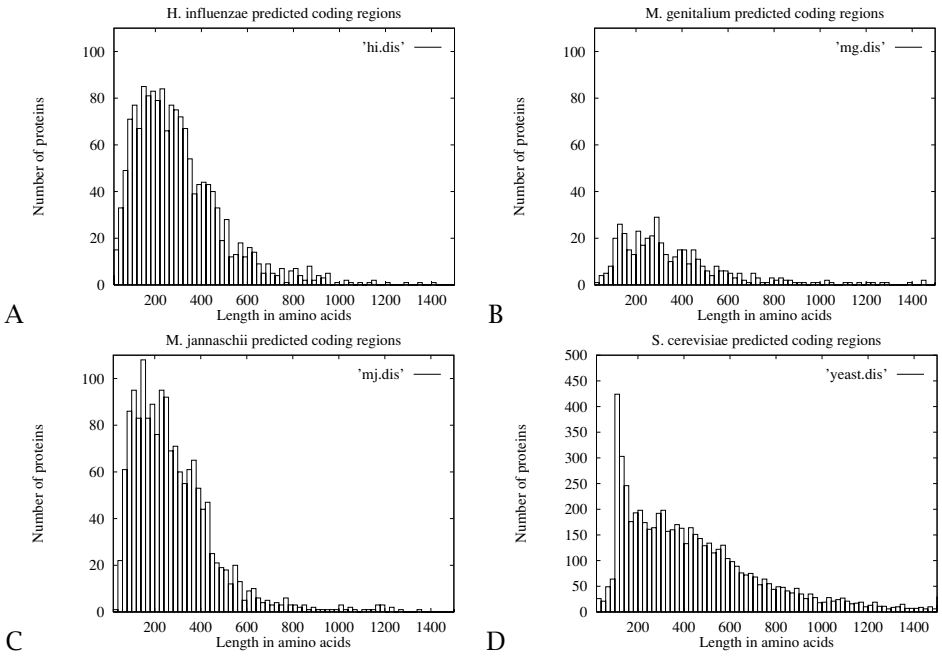

Figure 1.4: Length Distributions for Predicted Protein Coding Regions in Entire Genomes. **A.** *H. influenzae*, among the 1,743 regions, amino acid chains of lengths between 140 and 160 are the most frequent. **B.** *M. genitalium* with 468 regions, and preferred amino acid chains of length between 120 and 140 or 280 and 300. **C.** The archaeon *M. jannaschii* with 1,735 regions; amino acid chains of length between 140 and 160 are the most frequent. **D.** *S. cerevisiae*, among the 6,200 putative protein coding regions, amino acid chains of length between 100 and 120 are the most frequent; this interval is followed by the interval 120 to 140. As described in a 1997 correspondence in *Nature*, the *S. cerevisiae* set clearly contains an overrepresentation (of artifact sequences) in the 100–120 length interval [144].

Several other types of long-range regularities have been investigated, for example, the preference for identical or similar residue partners in beta-sheets [543, 570, 268, 45] and in close contact pairs [273], the long- and short-distance periodicity in packing density [175], and whether mutations in the amino acid sequence are significantly correlated over long distances [515, 485, 214].

The advent of the complete genomes from both prokaryotic and eukaryotic organisms has made it possible to check whether earlier observations based on incomplete and redundant data hold true when single organisms are compared. One quite surprising observation has been that proteins appear to be made out of different sequence units with characteristic length of ≈ 125 amino acids in eukaryotes and ≈ 150 amino acids in prokaryotes [64]. This indicates a

possible underlying order in protein sequence organization that is more fundamental than the sequence itself. If such a systematics has indeed been created by evolution, the *length* distributions of the polypeptide chains may be more fundamental than what conventionally is known as the "primary" structure of proteins.

In 1995 the first complete genome of a free living organism, the prokaryote *Haemophilus influenzae*, was published and made available for analysis [183]. This circular genome contains 1,830,137 bp with 1,743 predicted protein coding regions and 76 genes encoding RNA molecules. In figure 1.4 the length distribution of *all* the putative proteins in this organism is shown. For comparison, the figure also shows the length distributions of the $\approx$ 468 proteins in the complete *Mycoplasma genitalium* genome [189], as well as the $\approx$ 1,735 predicted protein coding regions in the complete genome of the archaeon *Methanococcus jannaschii* [105].

By comparing *Saccharomyces cerevisiae* (figure 1.4) against the distributions for the prokaryotes, it is possible by mere inspection to observe that the peaks for the prokaryote *H. influenzae* and the eukaryote *S. cerevisiae* are positioned in what clearly are different intervals: at 140–160 and 100–120, respectively.

Performing *redundancy* reduction together with spectral analysis has led to the conclusion that a eukaryotic distribution from a wide range of species peaks at 125 amino acids and that the distribution displays a periodicity based on this size unit [64]. Figure 1.4D also clearly shows that weaker secondary and tertiary peaks are present around 210 and 330 amino acids. This distribution is based on the entire set of proteins in this organism, and not a redundancy reduced version.

Interestingly, the distribution for the archaeon *M. jannaschii* lies in between the *H. influenzae* and the *S. cerevisiae* distributions. This is in accordance with the emerging view that the archaeon kingdom shares many similarities with eukaryotes rather than representing a special kind of bacteria in the prokaryotic kingdom [564, 105, 197]. This indicates that the universal ancestral progenote has induced conserved features in genomes of bacteria, archaea, and eucaryota:

$$\texttt{prokaryota}(\texttt{nonucleus}) \not\equiv \texttt{bacteria}. \tag{1.2}$$

This classification issue for archaeon organisms has led to confusion in textbooks and in the rational basis for classifying organisms in sequence databases [197].

Annotated protein primary structures also accumulate rapidly in the public databases. Table 1.4 shows the number of protein sequences in the top-scoring organisms in one of the protein sequence databases, SWISS-PROT [24]. Figure

Species	Sequences
Homo sapiens	6,742
Saccharomyces cerevisiae	4,845
Escherichia coli	4,661
Mus musculus	4,269
Rattus norvegicus	2,809
Bacillus subtilis	2,229
Caenorhabditis elegans	2,163
Haemophilus influenzae	1,746
Schizosaccharomyces pombe	1,654
Drosophila melanogaster	1,443
Methanococcus jannaschii	1,429
Arabidopsis thaliana	1,240
Mycobacterium tuberculosi	1,228
Bos bovis	1,202
Gallus gallus	948

Table 1.4: The Number of Sequences for the 15 Most Abundant Organisms in SWISS-PROT rel. 39.16, April 2001.

1.5 shows the development of the size of this database. Like GenBank, it grows exponentially, although at a much slower pace. This illustrates how much more slowly the biologically meaningful interpretation of the predicted genes arises. New techniques are needed, especially for functional annotation of the information stemming from the DNA sequencing projects [513].

Another database which grows even more slowly is the Protein Data Bank (PDB). This reflects naturally the amount of experimental effort that normally is associated with the determination of three dimensional protein structure, whether performed by X-ray crystallography or NMR. Still, as can be seen in Figure 1.6 this database also grows exponentially, and due to the initiation of many structural genomics projects in the US, Japan and Europe it is very likely that this pattern will continue for quite a while.

1.3.3 Protein Function

Many functional aspects of proteins are determined mainly by local sequence characteristics, and do not depend critically on a full 3D structure maintained in part by long-range interactions [149]. In the context of overall functional prediction, these characteristics can provide essential hints toward the precise function of a particular protein, but they can also be of significant value in establishing negative conclusions regarding compartmentalization—for example, that a given protein is nonsecretory or nonnuclear.

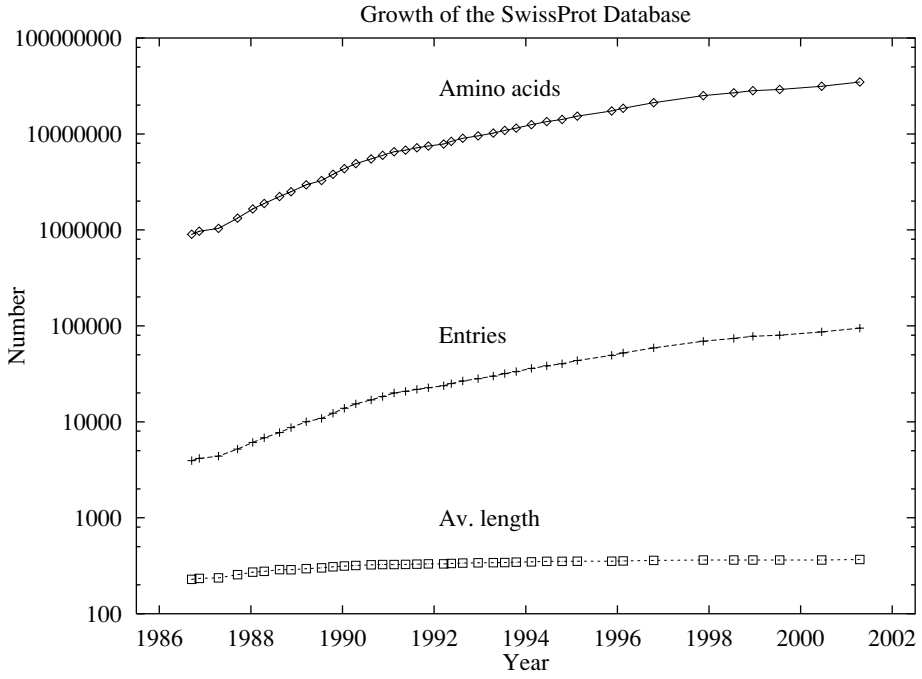

Figure 1.5: The Exponential Growth of the SWISS-PROT Database in the Period 1987–2001. The size of SWISS-PROT rel. 39.16 is in the order of 34,800,000 amino acids from 95,000 entries.

One of the major tasks within bioinformatics in the postgenome era will be to find out what the genes really do in concerted action, either by simultaneous measurement of the activity of arrays of genes or by analyzing the cell's protein complement [408, 360, 413]. It is not unlikely that it will be hard to determine the function of many proteins experimentally, because the function may be related specifically to the native environment in which a particular organism lives. Bakers yeast, *Saccharomyces cerevisiae*, has not by evolution been designed for the purpose of baking bread, but has been shaped to fit as a habitant of plant crops like grapes and figs [215]. Many genes may be included in the genome for the purpose of securing survival in a particular environment, and may have no use in the artificial environment created in the laboratory. It may even, in many cases, be almost impossible to imitate the natural host, with its myriad other microorganisms, and thereby determine the exact function of a gene or gene product by experiment.

The only effective route toward the elucidation of the function of some of

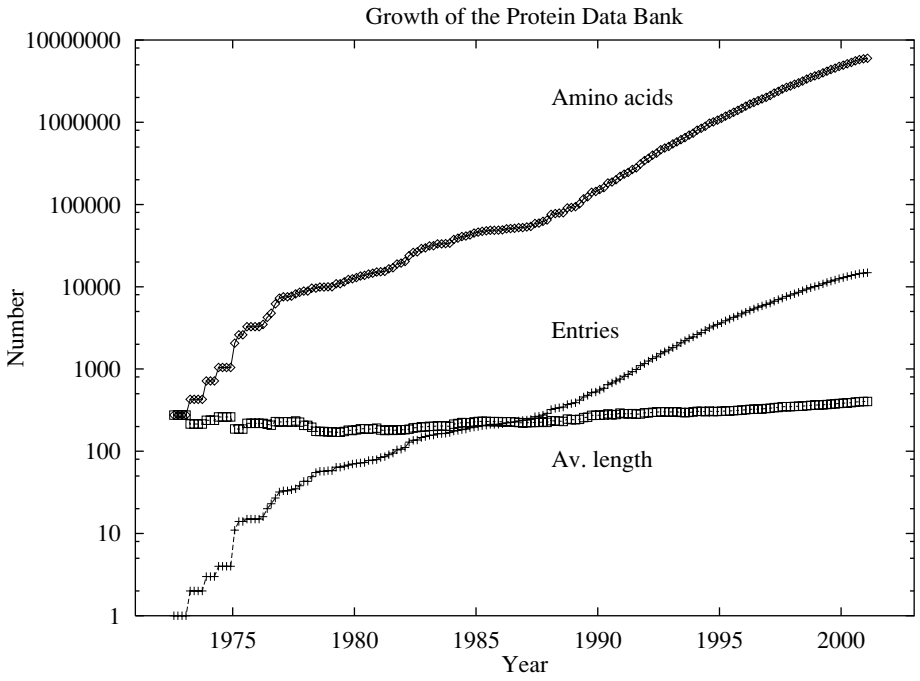

Figure 1.6: The Exponential Growth of the PDB Database in the Period 1972–2001. The size of PDB (April 19, 2001) is in the order of 6,033,000 amino acids from 14,910 entries (average length 405 aa).

these so-called orphan proteins may be computational analysis and prediction, which can produce valuable indirect evidence for their function. Many protein characteristics can be inferred from the sequence. Some sequence features will be related to cotranslational or postfolding modifications; others, to structural regions providing evidence for a particular general three-dimensional topology. Prediction along these lines will give a first hint toward functionality that later can be subjected to experimental verification [288].

In the last couple of years a number of methods that do not rely on direct sequence similarity have been published [380, 162, 271, 378]. One quite successful method has been exploiting gene expression data obtained using DNA array [425] and chip technology (see chapter 12). Genes of unknown function that belong to a cluster of genes displaying similar expression over time, or tissue types, may be assigned the function of the most prevalent gene function in that cluster (provided the cluster has genes with known function as members).

In this way functional information may be transferred between genes with little or no sequence similarity. However, coregulated genes may also in many cases have widely different functions, so often this approach cannot be used alone. Another problem is that as the DNA arrays become larger and larger, covering for example an entire mammalian genome, more and more clusters of genes significantly down- or upregulated will appear, where not a single gene has functional information assigned to it.

Another approach is the so-called "Rosetta stone" method, which is based on patterns of domain fusions [379, 167]. The underlying idea is that if two proteins in one organism exist as one fused multidomain protein in another organism, this may indicate that the two proteins are involved in performing the same function even though they are not directly related in sequence.

A third tool that can be used for linking together proteins of similar function is phylogenetic profiles [423]. In phylogenetic profiles each protein is represented as the organisms in which homologs are observed. If two proteins have identical (or very similar) phylogenetic profiles it indicates that they normally are observed together—an organism encodes either both or neither of the proteins in its genome. One possible explanation for this is that the proteins together perform a similar function. Phylogenetic profiles should be expected to become more powerful as more genomes become available. They have been successfully applied to the yeast genome but until several multicellular organisms have been sequenced they are of limited use for predicting the function of human proteins.

1.3.4 Protein Function and Gene Ontologies

Genomewide assignment of function requires that the functional role of proteins be described in a systematic manner using well defined categories, keywords, and hierachies. A gene ontology is essentially a specification of relevant concepts in molecular biology and the relationships among those concepts. If information in the scientific literature and in databases is to be shared in the most useful way, ontologies must be exchanged in a form that uses standardized syntax and semantics. In practice this means for example that functional categories and systematics must be designed to cover a wide range of organisms, if not all, and that the system is able to incorporate new discoveries as they appear over time.

One of the major developments [21, 22] in this area has been the creation of the *Gene Ontology Consortium*, which has participation from different areas, including fruitfly (FlyBase), budding yeast (Saccharomyces Genome Database), mouse (Mouse Genome and Gene Expression Databases), brassica (The Arabidopsis Information Resource), and nematode (WormBase). The goal of the

Gene Ontology Consortium is to produce a dynamic controlled vocabulary that is based on three organizing principles and functional aspects: (1) molecular function, (2) biological process and (3) cellular component. A protein can represent one or more molecular functions, be used in one or more biological processes, and be associated with one or more cellular components.

Molecular function describes the tasks performed by individual gene products; examples are transcription factor and DNA helicase. Biological process describes broad biological goals, such as mitosis or purine metabolism, that are accomplished by ordered assemblies of molecular functions. Cellular component encompasses subcellular structures, locations, and macromolecular complexes; examples include nucleus, telomere, and origin recognition complex.

There are many ways to construct ontologies, including some with focus on molecular complexes or the immune system; see for example the RiboWeb ontology [123] or the ImMunoGenetics ontology [213]. Another prominent example is the *EcoCyc* ontology [307, 308], which is the ontology used in a database describing the genome and the biochemical machinery of *E. coli.* The database describes pathways, reactions, and enzymes of a variety of organisms, with a microbial focus. *EcoCyc* describes for example each metabolic enzyme of *E. coli*, including its cofactors, activators, inhibitors, and subunit structure. When known, the genes encoding the subunits of an enzyme are also listed, as well as the map position of a gene on the *E. coli* chromosome.

1.4 On the Information Content of Biological Sequences

The concept of information and its quantification is essential for understanding the basic principles of machine-learning approaches in molecular biology (for basic definitions see appendix B, for a review see [577]). Data-driven prediction methods should be able to extract essential features from individual examples and to discard unwanted information when present. These methods should be able to distinguish positive cases from negative ones, also in the common situation where a huge excess of negative, nonfunctional sites and regions are present in a genome. This discrimination problem is of course intimately related to the molecular recognition problem [363, 544, 474] in the cellular environment: How can macromolecules find the sites they are supposed to interact with when similar sites are present in very large numbers?

Machine-learning techniques are excellent for the task of discarding and compacting redundant sequence information. A neural network will, if not unreasonably oversized, use its adjustable parameters for storing common features that apply to many data items, and not allocate individual parameters to individual sequence patterns. The encoding principle behind neural network

training procedures superimposes sequences upon one another in a way that transforms a complex topology in the input sequence space into a simpler representation. In this representation, related functional or structural categories end up clustered rather than scattered, as they often are in sequence space.

For example, the set of all amino acid segments of length 13, where the central residue is in a helical conformation, is scattered over a very large part of the sequence space of segments of length 13. The same holds true for other types of protein secondary structures like sheets and turns. In this sequence space, 20^{13} possible segments exist (when excluding the twenty-first amino acid, selenocysteine). The different structural categories are typically *not* found in nicely separated regions of sequence space [297, 244]; rather, islands of sheets are found in sequence regions where segments preferably adopt a helical conformation, and vice versa. Machine-learning techniques are used because of their ability to cope with nonlinearities and to find more complex correlations in sequence spaces that are not functionally segregated into continuous domains.

Some sequence segments may even have ability to attain both the helix and the sheet conformation, depending on the past history of interaction with other macromolecules and the environment. Notably, this may be the case for the prion proteins, which recently have been associated with mad cow disease, and in humans with the Creutzfeldt–Jakob syndrome. In these proteins the same sequence may adopt different very stable conformations: a normal conformation comprising a bundle of helices and a disease-inducing "bad" conformation with a mixture of helices and sheets. The bad-conformation prions even have an autocatalytic effect, and can be responsible for the transformation of normal conformation prions into bad ones [266, 267, 444]. In effect, the protein itself serves as carrier of structural information which can be inherited. To distinguish this pathogen from conventional genetic material, the term "prion" was introduced to emphasize its proteinaceous and infectious nature. The 1997 Nobel Prize for Physiology or Medicine was given to Stanley B. Prusiner for his work on prions. The proposal that proteins alone can transmit an infectious disease has come as a considerable surprise to the scientific community, and the mechanisms underlying their function remain a matter of hot debate.

Based on local sequence information, such conformational conflicts as those in the prion proteins will of course be impossible to settle by any prediction method. However, a local method may be able to report that a piece of sequence may have a higher potential for, say, both helix and sheet as opposed to coil. This has actually been the case for the prion sequences [266, 267] when they are analyzed by one of the very successful machine-learning methods in sequence analysis, the PHD method of Rost and Sander. We return to this and other methods for the prediction of protein secondary

structure in chapter 6.

Another issue related to redundancy is the relative importance of individual amino acids in specifying the tertiary structure of a protein [347]. To put it differently: What fraction of a protein's amino acid sequence is sufficient to specify its structure? A prize—the Paracelsus Challenge—has even been put forth to stimulate research into the role of sequence specificity in contrast to protein stability [450, 291, 449]. The task is to convert one protein fold into another, while retaining 50% of the original sequence. Recently, a protein that is predominantly beta-sheet has in this way been transmuted into a native-like, stable, four-helix bundle [143]. These studies clearly show that the residues determine the fold in a highly nonlinear manner. The identification of the minimal requirements to specify a given fold will not only be important for the design of prediction approaches, but also a significant step towards solving the protein folding problem [143].

The analysis of the redundancy and information content of biological sequences has been strongly influenced by linguistics since the late 1950s. Molecular biology came to life at a time when scientific methodology in general was affected by linguistic philosophy [326]. Many influential ideas stemming from the philosophical and mathematical treatment of natural languages were for that reason partly "recycled" for the analysis of "natural" biological sequences—and still are for that matter (see chapter 11). The digital nature of genetic information and the fact that biological sequences are translated from one representation to another in several consecutive steps have also contributed strongly to the links and analogies between the two subjects.

The study of the translation genetic code itself was similarly influenced by the time at which the code was cracked. The assignment of the 20 amino acids and the translation stop signal to the 64 codon triplets took place in the 1960s, when the most essential feature a code could have was its ability to perform error correction. At that time the recovery of messages from spacecraft was a key topic in coding and information theory. Shannon's information-theoretical procedures for the use of redundancy in encoding to transmit data over noisy channels without loss were in focus. In the case of the genetic code, its block structure ensures that the most frequent errors in the codon–anticodon recognition will produce either the same amino acid, as intended, or insert an amino acid with at least some similar physicochemical properties, most notably its hydrophobicity. The importance of other nonerror-correcting properties of the genetic code may have been underestimated, and we shall see in chapter 6 that a neural network trained on the mapping between nucleotide triplets and amino acids is simpler for the standard code, and much more complex when trained on more error-correcting genetic codes that have been suggested as potential alternatives to the code found by evolution [524].

The amount of information in biological sequences is related to their com-

pressibility. Intuitively, simple sequences with many repeats can be represented using a shorter description than complex and random sequences that never repeat themselves. Data-compression algorithms are commonly used in computers for increasing the capacity of disks, CD-ROMs, and magnetic tapes. Conventional text-compression schemes are so constructed that they can recover the original data perfectly without losing a single bit. Text-compression algorithms are designed to provide a shorter description in the form of a less redundant representation—normally called a code—that may be interpreted and converted back into the uncompressed message in a reversible manner [447]. The literature on molecular biology itself is full of such code words, which shortens this particular type of text. The abbreviation DNA, for *deoxyribonucleic acid*, is one example that contributes to the compression of this book [577].

In some text sequences—for example, the source code of a computer program—losing a symbol may change its meaning drastically, while compressed representations of other types of data may be useful even if the original message cannot be recovered completely. One common example is sound data. When sound data is transmitted over telephone lines, it is less critical to reproduce everything, so "lossy" decompression in this case can be acceptable. In lossless compression, the encoded version is a kind of program for computing the original data. In later chapters both implicit and explicit use of compression in connection with machine learning will be described.

In section 1.2 an experimental approach to the analysis of the redundancy of large genomes was described. If large genomes contained just a proportionally increased number of copies of each gene, the kinetics of DNA renaturation experiments would be much faster than observed. Therefore, the extra DNA in voluminous genomes most likely does not code for proteins [116], and consequently algorithmic compression of sequence data becomes a less trivial task.

The study of the statistical properties of repeated segments in biological sequences, and especially their relation to the evolution of genomes, is highly informative. Such analysis provides much evidence for events more complex than the fixation and incorporation of single stochastically generated mutations. Combination of interacting genomes, both between individuals in the same species and by horizontal transfer of genetic information between species, represents intergenome communication, which makes the analysis of evolutionary pathways difficult.

Nature makes seemingly wasteful and extravagant combinations of genomes that become sterile organisms unable to contribute further to the evolution of the gene pool. Mules are well-known sterile crosses of horses and donkeys. Less well known are *ligers*, the offspring of mating male LIons and female tiGERS. *Tigrons* also exist. In contrast to their parents, they are very nervous and uneasy animals; visually they are true blends of the most char-

Figure 1.7: A Photograph of a Liger, the Cross between a Lion and a Tiger. Courtesy of the Los Angeles Wild Animal Way Station (Beverly Setlowe).

acteristic features of lions and tigers. It is unclear whether free-living ligers can be found in the wild; most of their potential parents inhabit different continents[1], but at the Los Angeles Wild Animal Way Station several ligers have been placed by private owners who could no longer keep them on their premises. Figure 1.7 shows this fascinating and intriguing animal.

In biological sequences *repeats* are clearly—from a description length viewpoint—good targets for compaction. Even in naturally occurring sequence without repeats, the statistical biases—for example, skew dipeptide, and skew di- and trinucleotide, distributions—will make it possible to find shorter symbol sequences where the original message can be rewritten using representative words and extended alphabets.

The ratio between the size of an encoded corpus of sequences and the original corpus of sequences yields the compression rate, which quantifies

[1] In a few Asian regions, lions and tigers live close to one another, for example, in Gujarat in the northwestern part of India.

globally the degree of regularity in the data:

$$R_C = \frac{S_E}{S_O}. \tag{1.3}$$

One important difference between natural text and DNA is that repeats occur differently. In long natural texts, repeats are often quite small and close to each other, while in DNA, long repeats can be found far from each other [447]. This makes conventional sequential compression schemes [56] less effective on DNA and protein data. Still, significant compression can be obtained even by algorithms designed for other types of data, for example, the `compress` routine from the UNIX environment, which is based on the Lempel–Ziv algorithm [551]. Not surprisingly, coding regions, with their reading frame and triplet regularity, will normally be more compressible than more random noncoding regions like introns [279]. Functional RNAs are in general considered to be less repetitive than most other sequences [326], but their high potential for folding into secondary structures gives them another kind of inherent structure, reducing their randomness or information content.

Hidden Markov models are powerful means for analyzing the sequential pattern of monomers in sequences [154]. They are generative models that can produce any possible sequence in a given language, each message with its own probability. Since the models normally are trained to embody the regularity in a sequence set, the vast majority of possible sequences end up having a probability very close to 0. If the training is successful, the sequences in the training set (and, hopefully, their homologues) end up having a higher probability. One may think of a hidden Markov model as a tool for parameterizing a distribution over the space of all possible sequences on a given alphabet. A particular family of proteins—globins, for example—will be a cloud of points in sequence space. Training a model on some of these sequences is an attempt to create a distribution on sequence space that is peaked over that cloud.

1.4.1 Information and Information Reduction

Classification and prediction algorithms are in general computational means for *reducing* the amount of information. The input is information-rich sequence data, and the output may be a single number or, in the simplest case, a *yes* or *no* representing a choice between two categories. In the latter case the output holds a maximum of one bit if the two possibilities are equally likely. A segregation of amino acid residues, depending on whether they are in an alpha-helical conformation or not, will be such a dichotomy, where the average output information will be significantly under one bit per residue, because in natural proteins roughly only 30% of the amino acids are found in the heli-

cal category. On average less than one yes/no question will then be required to "guess" the conformational categories along the sequences.

The contractive character of these algorithms means that they cannot be inverted; prediction programs cannot be executed backward and thus return the input information. From the output of a neural network that predicts the structural class of an amino acid residue, one cannot tell what particular input amino acid it was, and even less its context of other residues. Similarly, the log-likelihood from a hidden Markov model will not make it possible to reproduce the original sequence to any degree.

In general, computation discards information and proceeds in a logically irreversible fashion. This is true even for simple addition of numbers; the sum does not hold information of the values of the addends. This is also true for much of the sequence-related information processing that takes place in the cell. The genetic code itself provides a most prominent example: the degenerate mapping between the 64 triplets and the 20 amino acids plus the translation stop signal. For all but two amino acids, methionine and tryptophan, the choice between several triplets will make it impossible to retrieve the encoding mRNA sequence from the amino acids in the protein or which of the three possible stop codons actually terminated the translation. The individual probability distribution over the triplets in a given organism—known as its codon usage—determines how much information the translation will discard in practice.

Another very important example is the preceding process, which in eukaryotes produces the mature mRNA from the pre-mRNA transcript of the genomic DNA. The noncoding regions, introns, which interrupt the protein coding part, are removed and spliced out in the cell nucleus (see also sections 1.1.2 and 6.5.4) But from the mature mRNA it seems difficult or impossible to locate with high precision the junctions where the intervening sequences belonged [495, 496], and it will surely be impossible to reproduce the intron sequence from the spliced transcript. Most of the conserved local information at the splice junctions is in the introns. This makes sense because the exons, making up the mature mRNA sequence, then are unconstrained in terms of their protein encoding potential. Interestingly, specific proteins seem to associate with the exon-exon junctions in the mature mRNA only as a consequence of splicing [256], thus making the spliced messenger "remember" where the introns were. The splicing machinery leaves behind such signature proteins at the junctions, perhaps with the purpose of influencing downstream metabolic events in vivo such as mRNA transport, decay and translation.

Among the more exotic examples of clear-cut information reduction are phenomena like RNA editing [59] and the removal of "inteins" from proteins [301, 257]. In RNA editing the original transcript is postprocessed using guide RNA sequences found elsewhere in the genome. Either single nucleotides or

longer pieces are changed or skipped. It is clear that the original RNA copy of the gene cannot in any way be recovered from the edited mRNA.

It has also been discovered that polypeptide chains in some cases are spliced, sequence fragments known as inteins are removed, and the chain ends are subsequently ligated together. In the complete genome of the archaeon *Methanococcus jannaschii*, a surprisingly large number of inteins were discovered in the predicted open reading frames. Many other examples of logically and physically irreversible processes exist. This fact is of course related to the irreversible thermodynamic nature of most life processes.

The information reduction inherent in computational classification and prediction makes it easier to see why in general it does not help to add extra input data to a method processing a single data item. If strong and valuable correlations are not present in the extra data added, the method is given the increased burden of discarding even more information on the route toward the output of a single bit or two. Despite the fact that the extra data contain some exploitable features, the result will often be a lower signal-to-noise level and a decreased prediction performance (see chapter 6).

Protein secondary structure prediction normally works better when based on 13 amino acid segments instead of segments of size 23 or higher. This is not due solely to the curse of dimensionality of the input space, with a more sparse coverage of the space by a fixed number of examples [70]. Given the amount of three-dimensional protein structure data that we have, the amount of noise in the context of 10 extra residues exceeds the positive effect of the long-range correlations that are in fact present in the additional sequence data.

Machine-learning approaches may have some advantages over other methods in having a built-in robustness when presented with uncorrelated data features. Weights in neural networks vanish during training unless positive or negative correlations keep them alive and put them into use. This means that the 23-amino-acid context not will be a catastrophe; but it still cannot outperform a method designed to handle an input space where the relation between signal and noise is more balanced.

Information reduction is a key feature in the *understanding* of almost any kind of system. As described above, a machine-learning algorithm will create a simpler representation of a sequence space that can be much more powerful and useful than the original data containing all details.

The author of *Alice in Wonderland*, the mathematician Charles Dodgson (Lewis Carroll), more than 100 years ago wrote about practical issues in relation to maps and mapping. In the story “Sylvie and Bruno Concluded” the character Mein Herr tells about the most profound map one can think of, a map with the scale *one kilometer per kilometer*. He is asked, “Have you used it much?” He answers, “It has not been unfolded yet. The farmers were against it. They said that it would cover all the soil and keep the sunlight out! Now we

use the country itself, as its own map. And I can assure you that it is almost as good."

In the perspective of Mein Herr, we should stay with the unstructured, flat-file public databases as they are, and not try to enhance the principal features by using neural networks or hidden Markov models.

1.4.2 Alignment Versus Prediction: When Are Alignments Reliable?

In order to obtain additional functional insights as well as additional hints toward structural and functional relationships, new sequences are normally aligned against all sequences in a number of large databases [79]. The fundamental question is: When is the sequence similarity high enough that one may safely infer either a structural or a functional similarity from the pairwise alignment of two sequences? In other words, given that the alignment method has detected an overlap in a sequence segment, can a similarity threshold be defined that sifts out cases where the inference will be reliable? Below the threshold some pairs will be related and some will not, so subthreshold matches cannot be used to obtain negative conclusions. It is well known that proteins can be structurally very similar even if the sequence similarity is very low. At such low similarity levels, pure chance will produce other pairwise alignments that will mix with those produced by genuinely related pairs.

The nontrivial answer to this question is that it depends entirely on the particular structural or functional aspect one wants to investigate. The necessary and sufficient similarity threshold will be different for each task. Safe structural inference will demand a similarity at one level, and functional inference will in general require a new threshold for each functional aspect. Functional aspects may be related to a sequence as a whole—for example, whether or not a sequence belongs to a given class of enzymes. Many other functional aspects depend entirely on the local sequence composition. For example, does the N-terminal of a protein sequence have a signal peptide cleavage site at a given position or not?

In general, one may say that in the zone of safe inference, alignment should be preferred to prediction. In the best situations, prediction methods should enlarge the regions of safe inference. This can be done by evaluation of the confidence levels that are produced along with the predictions from many methods, a theme treated in more detail in chapter 5.

Sander and Schneider pioneered the algorithmic investigation of the relationship between protein sequence similarity and structural similarity [462]. In a plot of the alignment length against the percentage of identical residues in the overlap, two domains could be discerned: one of exclusively structurally similar pairs, and one containing a mixture of similar and nonsimilar pairs.

Structural similarity was defined by more than 70% secondary structure assignment identity in the overlap. It was observed that this criterion corresponds to a maximum root-mean-square deviation of 2.5Å for a structural alignment of the two fragments in three dimensions. The mixed region reflects the fact that the secondary structure identity may exceed 70% by chance, especially for very short overlaps, even in pairs of completely unrelated sequences.

The border between the two domains, and thereby the threshold for sequence similarity, measured in percentage identity, depends on the length of the aligned region (the overlap). Sander and Schneider defined a length-dependent threshold function: for overlap length $l < 10$, no pairs are above the threshold; for $10 < l < 80$, the threshold is $290.15l^{-0.562}$%; and for $l > 80$, the threshold is 24.8%.

This threshold can be used to answer the question whether alignment is likely to lead to a reliable inference, or whether one is forced to look for prediction methods that may be available for the particular task. *If* the new sequence is superthreshold, alignment or homology building should be the preferred approach; if it is subthreshold, prediction approaches by more advanced pattern-recognition schemes should be employed, possibly in concert with the alignment methods.

In this type of analysis the "safe zone of inference" is of course not 100% *safe* and should be used as a guideline only, for example when constructing test sets for validation of high-throughput prediction algorithms. In many cases the change of a single amino acid is known to lead to a completely different, possibly unfolded and unfunctional protein. Part of the goal in the so-called *single-nucleotide polymorphism* projects is to identify coding SNPs, which may affect protein conformation and thereby for example influence disease susceptibility and/or alter the effect of drugs interacting with a particular protein [394].

1.4.3 Prediction of Functional Features

The sequence identity threshold for structural problems cannot be used directly in sequence prediction problems involving *functionality*. If the aim is safe inference of the exact position of a signal peptide cleavage site in a new sequence from experimentally verified sites in sequences from a database, it is a priori completely unknown what the required degree of similarity should be.

Above, "structurally similar" was defined by quantification of the mean distance in space. In an alignment, *functional* similarity means that any two residues with similar function should match without any shift. Two cleavage sites should line up exactly residue by residue, if a site in one sequence should

be positioned unambiguously by the site in the other. In practice, whether a perfect separation between a completely safe zone and a mixed zone can be obtained by alignment alone will depend on the degree of conservation of different types of functional sites.

This binary criterion for successful alignment can, together with a definition of the zone-separating principle, be used to determine a threshold function that gives the best discrimination of functional similarity [405]. The principle for establishing a nonarbitrary threshold is general; the approach may easily be generalized to other types of sequence analysis problems involving, for instance, glycosylation sites, phosphorylation sites, transit peptides for chloroplasts and mitochondria, or cleavage sites of polyproteins, and to nucleotide sequence analysis problems such as intron splice sites in pre-mRNA, ribosome binding sites, and promoters. But for each case a specific threshold must be determined.

For problems such as those involving splice sites in pre-mRNA or glycosylation sites of proteins, there are several sites per sequence. One way of addressing this problem is to split each sequence into a number of subsequences, one for each potential site, and then use the approach on the collection of subsequences. Alternatively, the fraction of aligned sites per alignment may be used as a functional similarity measure, in analogy with the structural similarity used by Sander and Schneider (the percentage of identical secondary structure assignments in the alignment). In this case, a threshold value for functional similarity—analogous to the 70% structural similarity threshold used by Sander and Schneider—must be defined before the similarity threshold can be calculated.

1.4.4 Global and Local Alignments and Substitution Matrix Entropies

The optimality of pairwise alignments between two sequences is not given by some canonical or unique criterion with universal applicability throughout the entire domain of sequences. The matches produced by alignment algorithms depend entirely on the parameters quantitatively defining the similarity of corresponding monomers, the cost of gaps and deletions, and most notably whether the algorithms are designed to optimize a score globally or locally.

Some problems of biological relevance concern an overall, or global, comparison between two sequences, possibly with the exception of the sequence ends, while others would be meaningless unless attacked by a subsequence angle for the localization of segments or sites with similar sequential structure.

Classical alignment algorithms are based on dynamic programming—for optimal global alignments, the Needleman–Wunsch algorithm [401, 481], and for optimal local alignments, the Smith–Waterman algorithm [492] (see chapter

4). Dynamic programming is a computing procedure to manage the combinatorial explosion that would result from an exhaustive evaluation of the scores associated with any conceivable alignment of two sequences. Still, dynamic programming is computationally demanding, and a number of heuristics for further reduction of the resources needed for finding significant alignments have been developed [417, 419]. Other very fast and reliable heuristic schemes do not build on dynamic programming, but interactively extend small subsequences into longer matches [13, 14]. The conventional alignment schemes have been described in detail elsewhere [550, 428]; here we will focus on some of the important aspects related to the preparation of dedicated data sets.

How "local" a local alignment scheme will be in practice is strongly influenced by the choice of substitution matrix. If the score level for matches is much higher than the penalty for mismatches, even local alignment schemes will tend to produce relatively long alignments. If the mismatch score will quickly eat up the match score, short, compact overlaps will result.

A substitution matrix specifies a set of scores s_{ij} for replacing amino acid i by amino acid j. Some matrices are generated from a simplified protein evolution model involving amino acid frequencies, p_i, and pairwise substitution frequencies, q_{ij}, observed in existing alignments of naturally occurring proteins. A match involving a rare amino acid may count more than a match involving a common amino acid, while a mismatch between two interchangeable amino acids contributes a higher score than a mismatch between two functionally unrelated amino acids. A mismatch with a nonnegative score is known as a similarity or a conservative replacement. Other types of substitution matrices are based on the relationships between the amino acids according to the genetic code, or physicochemical properties of amino acids, or simply whether amino acids in alignments are identical or not.

All these different substitution matrices can be compared and brought on an equal footing by the concept of substitution matrix entropy. As shown by Altschul [8], any amino acid substitution matrix is, either implicitly or explicitly, a matrix of logarithms of normalized target frequencies, since the substitution scores may be written as

$$s_{ij} = \frac{1}{\lambda}\left(\ln \frac{q_{ij}}{p_i p_j}\right) \tag{1.4}$$

where λ is a scaling factor. Changing λ will change the absolute value of the scores, but not the relative scores of different local alignments, so it will not affect the alignments [405].

The simplest possible scoring matrices are *identity matrices*, where all the diagonal elements have the same positive value (the match score, s), and all the off-diagonal elements have the same negative value (the mismatch score,

$\bar{s}$). This special case has been treated by Nielsen [405]. An identity matrix may be derived from the simplest possible model for amino acid substitutions, where all 20 amino acids appear with equal probability and the off-diagonal substitution frequencies are equal:

$$\begin{aligned} p_i &= \tfrac{1}{20} \quad \text{for all } i, \\ q_{ij} &= \begin{cases} q & \text{for } i = j \\ \bar{q} & \text{for } i \neq j. \end{cases} \end{aligned} \tag{1.5}$$

In other words, when an amino acid mutates, it has equal probabilities $\bar{q}$ of changing into any of the 19 other amino acids.

There is a range of different identity matrices, depending on the ratio between the positive and negative scores, $s/\bar{s}$. If $s = -\bar{s}$, a local alignment must necessarily contain more matches than mismatches in order to yield a positive score, resulting in short and strong alignments, while if $s \gg -\bar{s}$, one match can compensate for many mismatches, resulting in long and weak alignments. The percentage identity p in gapfree local identity matrix alignment has a minimum value

$$p > \frac{-\bar{s}}{s - \bar{s}}. \tag{1.6}$$

We define $r = \bar{q}/q$, the mutability or the probability that a given position in the sequence has changed into a random amino acid (including the original one). $r = 0$ corresponds to no changes, while $r = 1$ corresponds to an infinite evolutionary distance.

Since the sum of all q_{ij} must be 1, we use the relation $20q + 380\bar{q} = 1$ to calculate the target frequencies

$$q = \frac{1}{20 + 380r} \text{ and } \bar{q} = \frac{r}{20 + 380r} \tag{1.7}$$

and the s_{ij} values may be calculated using (1.4). Since the score ratio, $s/\bar{s}$, is independent of λ and therefore a function of r, we can calculate r numerically from the score ratio.

The *relative entropy* of an amino acid substitution matrix was defined thus by Altschul:

$$\mathcal{H} = \sum_{i,j} q_{ij} s_{ij} \text{bits} \tag{1.8}$$

where the s_{ij}s are normalized so that $\lambda = \ln 2$ (corresponding to using the base-2 logarithm in (1.4)). The relative entropy of a matrix can be interpreted as the amount of information carried by each position in the alignment (see also appendix B for all information-theoretic notions such as entropy and relative entropy).

The shorter the evolutionary distance assumed in the calculation of the matrix, the larger H is. At zero evolutionary distance ($r = 0$), the mismatch penalty $\bar{s}$ is infinite, that is, gaps are completely disallowed, and the relative entropy is equal to the entropy of the amino acid distribution: $\mathcal{H} = -\sum_i p_i \log_2 p_i$. In the identity model case, $\mathcal{H} = \log_2 20 \approx 4.32$ bits, and the local alignment problem is reduced to the problem of finding the longest common substring between two sequences. Conversely, as the evolutionary distance approaches infinity ($r \simeq 1$), all differences between the q_{ij} values disappear and $\mathcal{H}$ approaches 0.

1.4.5 Consensus Sequences and Sequence Logos

When studying the specificity of molecular binding sites, it has been common practice to create consensus sequences from alignments and then to choose the most common nucleotide or amino acid as representative at a given position [474]. Such a procedure throws a lot of information away, and it may be highly misleading when interpreted as a reliable assessment of the molecular specificity of the recognizing protein factors or nucleic acids. A somewhat better alternative is to observe all frequencies at all positions simultaneously.

A graphical visualization technique based on the Shannon information content at each position is the sequence *logo* approach developed by Schneider and coworkers [473]. The idea is to emphasize the deviation from the uniform, or flat, distribution, where all monomers occur with the same probability, p. In that case, $p = 0.25$ for nucleotide sequence alignments and $p = 0.05$ in amino acid sequence alignments.

Most functional sites display a significant degree of deviation from the flat distribution. From the observed frequencies of monomers at a given position, i, the deviation from the random case is computed by

$$D(i) = \log_2 |A| + \sum_{k=1}^{|A|} p_k(i) \log_2 p_k(i), \tag{1.9}$$

where $|A|$ is the length of the alphabet, normally 4 or 20. Since the logarithm used is base 2, $D(i)$ will be measured in bits per monomer. In an amino acid alignment $D(i)$ will be maximal and equal $\log_2 20 \approx 4.32$ when only one fully conserved amino acid is found at a given position. Similarly, the deviation will be two bits maximally in alignments of nucleotide sequences.

With the logo visualization technique a column of symbols is used to display the details of a consensus sequence. The total height of the column is equal to the value of $D(i)$, and the height of each monomer symbol, k, is proportional to its probability at that position, $p_k(i)$. Monomers drawn with different colors can be used to indicate physicochemical properties, such as

charge and hydrophobicity, or nucleotide interaction characteristics, such as weak or strong hydrogen bonding potential. Compared with the array of numbers in a weight matrix covering the alignment region, the logo technique is quite powerful and fairly easy to use. When D is summed over the region of the site, one gets a measure of the accumulated information in a given type of site, for example, a binding site. D may indicate the strength of a binding site, and can be compared against the information needed to find true sites in a complete genome or protein sequence [474]. With this information-theoretical formulation of the degree of sequence conservation, the problem of how proteins can find their required binding sites among a huge excess of nonsites can be addressed in a quantitative manner [474, 472].

Figures 1.8 and 1.9 show two examples of alignment frequencies visualized by the logo technique. The first is from an alignment of translation initiation sites in *E. coli*. In the nuclear part of eukaryotic genomes, the initiation triplet—the start codon—is very well conserved and is almost always AUG, representing the amino acid methionine. In prokaryotes several other initiation triplets occur with significant frequencies, and the logo shows to what extent the nucleotides at the three codon positions are conserved [422]. Since the same *E. coli* ribosome complex will recognize *all* translation initiation sites, the logo indicates the specificity of the interaction between ribosomal components and the triplet sequence. The conserved Shine–Dalgarno sequence immediately 5' to the initiation codon is used to position the mRNA on the ribosome by base pairing.

A logo is clearly most informative if only sequences that share a similar signal are included, but it can also be used in the process of identifying different patterns belonging to different parts of the data. In the extremely thermophilic archaeon *Sulfolobus solfataricus*, translation initiation patterns may depend on whether genes lie inside operons or at the start of an operon or single genes. In a recent study [523], a Shine-Dalgarno sequence was found upstream of the genes inside operons, but not for the first gene in an operon or isolated genes. This indicates that two different mechanisms are used for translation initiation in this organism.

Figure 1.9 displays a logo of mammalian amino acid sequence segments aligned at the start of alpha-helices [99]. The logo covers the transition region: to the left, the conformational categories of coil and turn appear most often, and to the right, amino acids frequent in alpha-helices are found at the tops of the columns. Interestingly, at the N-terminus, or the cap of the helix, the distribution of amino acids is more biased than within the helix itself [435]. A logo of the C-terminus helix shows the capping in the other end. Capping residues are presumably an integral part of this type of secondary structure, because their side chain hydrogen bonds stabilize the dipole of the helix [435]. An analogous delimitation of beta-sheets—so-called beta breakers—marks the

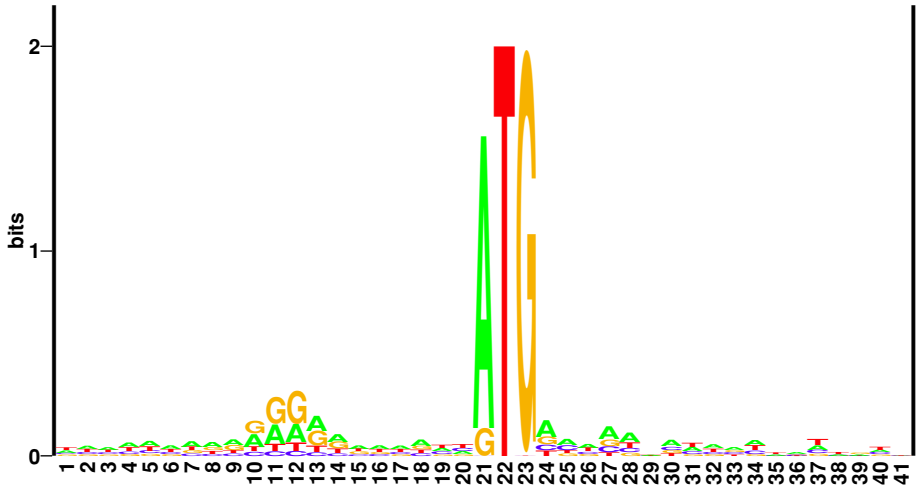

Figure 1.8: Logo Showing an Alignment of Translation Start Triplets That Are Active in *E. coli*. Translation starts at position 21 in the logo. The conventional initiation triplet ATG encoding methionine is by far the most abundant and dominates the logo. The data were obtained from [422].

termini of this chain topology [133].

Sequence logos are useful for a quick examination of the statistics in the context of functional sites or regions, but they can also show the range in which a sequence signal is present. If one aligns a large number of O-glycosylation sites and inspects the logo, the interval where the compositional bias extends will immediately be revealed. Such an analysis can be used not only to shape the architecture of a prediction method but also to consider what should actually be predicted. One may consider lumping O-glycosylated serines and threonines together if their context shares similar properties [235]. If they differ strongly, individual methods handling the two residue types separately should be considered instead. In the cellular environment, such a difference may also indicate that the enzymes that transfer the carbohydrates to the two residues are nonidentical.

Sequence logos using monomers will treat the positions in the context of a site *independently*. The logo will tell nothing about the correlation between the different positions, or whether the individual monomers appear simultaneously at a level beyond what would be expected from the single-position statistics. However, the visualization technique can easily handle the occurrence of, say, dinucleotides or dipeptides, and show pair correlations in the form of stacks of combined symbols. The size of the alphabets, $|A|$, in (1.9)

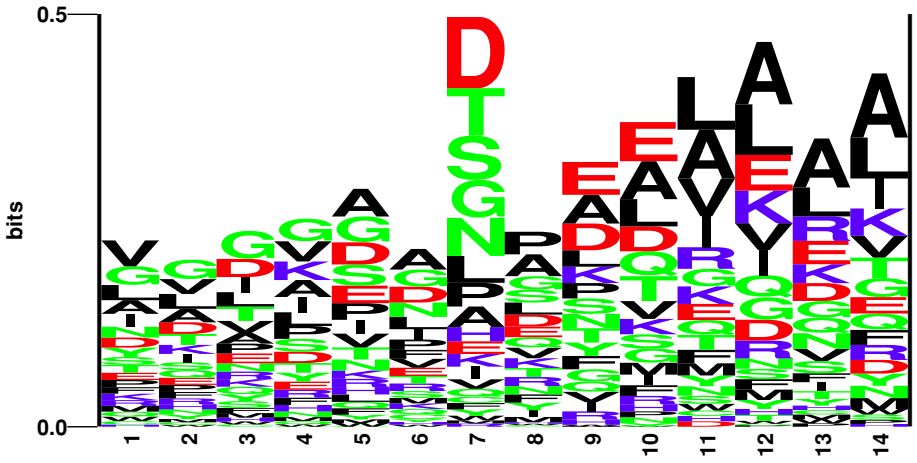

Figure 1.9: Logo Showing an Alignment of Alpha-Helix N-termini. The data comprised a nonredundant set of mammalian proteins with known three-dimensional structure [99]. The helix starts at position 7 in the logo. The secondary structure assignment was performed by the Kabsch and Sander algorithm [297]. The largest compositional bias in this region is observed at the position just before the helix start.

will change accordingly; otherwise, the same formula applies.

Figure 1.10 shows an example of a dinucleotide-based logo of donor splice sites in introns from the plant *Arabidopsis thaliana*. In addition to the well-known consensus dinucleotides GT and GC (almost invisible) at the splice junction in the center of the logo, the logo shows that the GT dinucleotide, which appears inside the intron at the third dinucleotide position, occurs more frequently than expected.

A slight variation of the logo formula (1.9), based on relative entropy (or Kullback–Leibler asymmetric divergence measure [342, 341]), is the following:

$$\mathcal{H}(i) = \mathcal{H}(P(i), Q(i)) = \sum_{k=1}^{|A|} p_k(i) \log \frac{p_k(i)}{q_k(i)}. \tag{1.10}$$

This quantifies the contrast between the observed probabilities $P(i)$ and a reference probability distribution $Q(i)$. Q may, or may not, depend on the position i in the alignment. When displaying the relative entropy, the height of each letter can also, as an alternative to the frequency itself, be computed from the background scaled frequency at that position [219].

In order for the logo to be a reliable description of the specificity, it is essential that the data entering the alignment be nonredundant. If a given

site is included in multiple copies, the probability distribution will be biased artificially.

In chapter 6 we will see how neural networks go beyond the positionwise uncorrelated analysis of the sequence, as is the case for the simple logo visualization technique and also for its weight matrix counterpart, where each position in the matrix is treated independently. A weight matrix assigns weights to the positions that are computed from the ratio of the frequencies of monomers in an alignment of some "positive" sites and the frequencies in a reference distribution. A sum of the logarithms of the weights assigned to given monomers in a particular sequence will give a score, and a threshold may be adjusted so that it will give the best recognition of true sites, in terms of either sensitivity or specificity.

Neural networks have the ability to process the sequence data nonlinearly where correlations between different positions can be taken into account. "Nonlinear" means essentially that the network will be able to produce correct predictions in cases where one category is correlated with one of two features, but not both of them simultaneously. A linear method would not be able to handle such a two-feature case correctly.

In more complex situations, many features may be present in a given type of site, with more complex patterns of correlation between them. The ability to handle such cases correctly by definition gives the neural network algorithms great power in the sequence data domain.

An O-glycosylation site may be one case where amino acids of both positive and negative charges may be acceptable and functional, but not both types at the same time. A conventional monomer weight matrix cannot handle this common situation. However, for some prediction problems one can get around the problem by developing weight matrices based on dipeptides or more complex input features. Another strategy may be to divide all the positive cases into two or more classes, each characterized by its own weight matrix. Such changes in the approach can in some cases effectively convert a nonlinear problem into a linear one.

In general, the drawback of linear techniques is that it becomes impossible to *subtract* evidence. In linear methods two types of evidence will combine and add up to produce a high score, even if the biological mechanism can accept only one of them at a time. A nonlinear method can avoid this situation simply by decreasing the score if the combined evidence from many features exceeds a certain level.

A clever change in the input representation will in many cases do part of the job of transforming the topology of the sequence space into a better-connected space in which isolated islands have been merged according to the functional class they belong to. Since the correlations and features in sequences often are largely unknown, at least when one starts the prediction analysis, the nonlinear

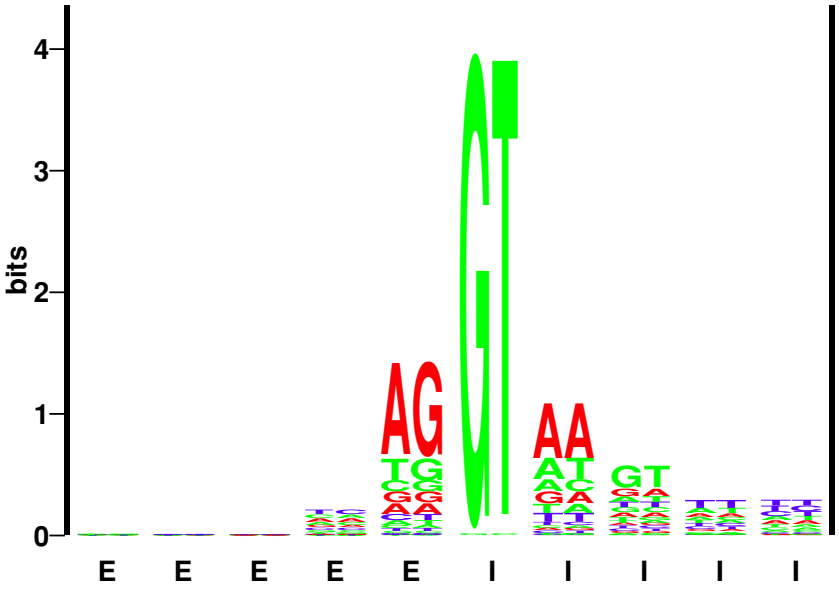

Figure 1.10: A Logo of Donor Splice Sites from the Dicot Plant *A. thaliana* (cress). The logo is based on frequencies of nonoverlapping dinucleotides in the exon/intron transition region, using the standard Shannon information measure entering equation (1.9) with the alphabet size $|A| = 16$. The logo was prepared on a nonredundant data set of sequences extracted from GenBank [327].

potential of neural networks gives them a big advantage in the development phase for many types of tasks.

The issue of which method to use has for many years been a highly dogmatic matter in artificial intelligence. In the data domain of biological sequences, it is clear that many different methods will be able to perform at the same level *if* one knows in advance which features to look for. If an analysis of the weights in a neural network trained on a given task (see chapter 6) shows that the network is being excited (or inhibited) toward a positive (or negative) prediction by specific sequence features, rules can often be constructed that also will have a high discriminatory power. It is the experience of many people that machine-learning methods are *productive* in the sense that near-optimal methods can be developed quite fast, given that the data are relatively clean; it often can be much harder to try to design powerful rules from scratch.

1.5 Prediction of Molecular Function and Structure

The methods and applications described in this book will be targeted toward the agenda formulated by von Heijne in his early book on sequence analysis: "What can you do with your sequence once you have it?" [540]. Applications well suited for treatment by machine-learning approaches will be described in detail in later chapters; here we give an annotated list of some important computational problems that have been tackled within this framework in the analysis of data from DNA, RNA, and protein sequences. In some cases sequences are represented by experimentally determined biochemical characteristics rather than symbols from a finite alphabet of monomers.

1.5.1 Sequence-based Analysis

In most cases, single-stranded sequences are used, no matter whether the object in the cellular environment is DNA or RNA. One exception is the analysis of structural elements of DNA, such as bendability or intrinsic bending potential, which must be based on a true double-stranded interpretation of the double helix.

Intron splice sites and branch points in eukaryotic pre-mRNA. Intervening sequences that interrupt the genes of RNA and proteins are characterized, but not unambiguously defined, by local features at the splice junctions. Introns in protein-encoding genes present the most significant computational challenge. In some organisms, nuclear introns are few and their splice sites are well conserved (as in *S. cerevisiae*), but in many other eukaryotes, including man, it is a major problem to locate the correct transition between coding and noncoding regions, and thus to determine the mature mRNA sequence from the genomic DNA. In yeast, introns occur mainly in genes encoding ribosomal proteins. The fact that genes in many organisms are being spliced differently, depending on tissue type or stage of development, complicates the task considerably. Weight matrices, neural networks, and hidden Markov models have been applied to this problem in a multitude of different versions.

Gene finding in prokaryotes and eukaryotes. Machine-learning techniques have been applied to almost all steps in computational gene finding, including the assignment of translation start and stop, quantification of reading frame potential, frame interruption of splice sites, exon assignment, gene modeling, and assembly. Usually, highly diverse combinations of machine-learning approaches have been incorporated in individual methods.

Recognition of promoters—transcription initiation and termination. Initiation of transcription is the first step in gene expression and constitutes an important point of control in the organism. The initiation event takes place when RNA polymerase—the enzyme that catalyzes production of RNA from the DNA template—recognizes and binds to certain DNA sequences called promoters. This prediction problem is hard due to both the large variable distance between various DNA signals that are the substrate for the recognition by the polymerase, and the many other factors involved in regulation of the expression level. The elastic matching abilities of hidden Markov models have made them ideal for this task, especially in eukaryotes, but neural networks with carefully designed input architecture have also been used.

Gene expression levels. This problem may be addressed by predicting the strength of known promoter signals if the expression levels associated with their genes have been determined experimentally. Alternatively, the expression level of genes may be predicted from the sequence of the coding sequence, where the codon usage and/or in some cases, the corresponding codon adaption indices, have been used to encode the sequence statistics.

Prediction of DNA bending and bendability. Many transactions are influenced and determined by the flexibility of the double helix. Transcription initiation is one of them, and prediction of transcription initiation or curvature/bendability from the sequence would therefore be valuable in the context of understanding a large range of DNA-related phenomena.

Nucleosome positioning signals. Intimately related to the DNA flexibility is the positioning of eukaryotic DNA when wrapped around the histone octamers in chromatin. Detection of the periodicity requires non-integer sensitivity—or an elastic matching ability as in hidden Markov models—because the signals occur every 10.1–10.6 bp, or every full turn of the double-stranded helix.

Sequence clustering and cluster topology. Because sequence data are notoriously redundant, it is important to have clustering techniques that will put sequences into groups, and also to estimate the intergroup distances at the same time. Both neural networks, in the form of self-organizing maps, and hidden Markov models have been very useful for doing this. One advantage over other clustering techniques has been the unproblematic treatment of large data sets comprising thousands of sequences.

Prediction of RNA secondary structure. The most powerful methods for computing and ranking potential secondary structures of mRNA, tRNA, and rRNA are based on the minimization of the free energy in the interaction be-

tween base pairs and between pairs of base pairs and their stacking energies [586, 260]. This is nontrivial for many reasons, one being that loop-to-loop interactions are hard to assess without a combinatorial explosion in the number of structures to be evaluated. Neural networks and grammar methods have had some success in handling features at which the more traditional minimization procedures for obtaining the energetically most favored conformation are less successful.

Other functional sites and classes of DNA and RNA. Many different types of sites have been considered for separate prediction, including branch points in introns, ribosome binding sites, motifs in protein–DNA interactions, other regulatory signals, DNA helix categories, restriction sites, DNA melting points, reading frame-interrupting deletions in EST sequences, classification of ribosomal RNAs according to phylogenetic classes, and classification of tRNA sequences according to species.

Protein structure prediction. This area has boosted the application of machine-learning techniques within sequence analysis, most notably through the work on prediction of protein secondary structure of Qian and Sejnowski [437]. Virtually all aspects of protein structure have been tackled by machine learning. Among the specific elements that have been predicted are categories of secondary structure, distance constraints between residues (contacts), fold class, secondary structure class or content, disulfide bridges between cysteine residues, family membership, helical transmembrane regions and topology of the membrane crossing, membrane protein class (number of transmembrane segments), MHC motifs, and solvent accessibility.

Protein function prediction. Functionally related features that have been considered for prediction are intracellular localization, signal peptide cleavage sites (secreted proteins), de novo design of signal peptide cleavage sites (optimized for cleavage efficiency), signal anchors (N-terminal parts of type-II membrane proteins), glycosylation signals for attachment of carbohydrates (the state and type of glycosylation determines the lifetime in circulation; this is strongly involved in recognition phenomena and sorting), phosphorylation and other modifications related to posttranslational modification (the presence of phosphorylation sites indicates that the protein is involved in intracellular signal transduction, cell cycle control, or mediating nutritional and environmental stress signals), various binding sites and active sites in proteins (enzymatic activity).

Protein family classification. The family association has been predicted from a global encoding of the dipeptide frequencies into self-organizing maps

and feed-forward neural networks, or local motif-based prediction that may enhance the detection of more distant family relationships.

Protein degradation. In all organisms proteins are degraded and recycled. In organisms with an immune system the specificity of the degradation is essential for its function and the successful discrimination between self and nonself. Different degradation pathways are active; in several of them proteins are unfolded prior to proteolytic cleavage, and therefore the specificity is presuambly strongly related to the pattern in the sequence and not to its 3D structure. This general problem has therefore quite naturally been attacked by machine-learning techniques, the main problem being the limited amount of experimentally characterized data.

Chapter 2

Machine-Learning Foundations: The Probabilistic Framework

2.1 Introduction: Bayesian Modeling

Machine learning is by and large a direct descendant of an older discipline, statistical model fitting. Like its predecessor, the goal in machine learning is to extract useful information from a corpus of data D by building good probabilistic models. The particular twist behind machine learning, however, is to automate this process as much as possible, often by using very flexible models characterized by large numbers of parameters, and to let the machine take care of the rest. Silicon machine learning also finds much of its inspiration in the learning abilities of its biological predecessor: the brain. Hence, a particular vocabulary is required in which "learning" often replaces "fitting."

Clearly, machine learning is driven by rapid technological progress in two areas:

- Sensors and storage devices that lead to large databases and data sets
- Computing power that makes possible the use of more complex models.

As pointed out in [455], machine-learning approaches are best suited for areas where there is a large amount of data but little theory. And this is exactly the situation in computational molecular biology.

While available sequence data are rapidly increasing, our current knowledge of biology constitutes only a small fraction of what remains to be discovered. Thus, in computational biology in particular, and more generally in biology and all other information-rich sciences, one must reason in the presence of a high degree of uncertainty: many facts are missing, and some of

the facts are wrong. Computational molecular biologists are, then, constantly faced with induction and *inference* problems: building models from available data. What are the right class of models and the right level of complexity? Which details are important and which should be discarded? How can one compare different models and select the best one, in light of available knowledge and sometimes limited data? In short, how do we know if a model is a good model? These questions are all the more acute in machine-learning approaches, because complex models, with several thousand parameters and more, are routinely used and sequence data, while often abundant, are inherently "noisy."

In situations where few data are available, the models used in machine-learning approaches have sometimes been criticized on the ground that they may be capable of accommodating almost any behavior for a certain setting of their parameters, and that simpler models with fewer parameters should be preferred to avoid overfitting. Machine-learning practitioners know that many *implicit* constraints emerge from the structure of the models and, in fact, render arbitrary behavior very difficult, if not impossible, to reproduce. More important, as pointed out in [397], choosing simpler models because few data are available does not make much sense. While it is a widespread practice and occasionally a useful heuristic, it is clear that the amount of data collected and the complexity of the underlying source are two completely different things. It is not hard to imagine situations characterized by a very complex source and relatively few data. Thus we contend that even in situations where data are scarce, machine-learning approaches should not be ruled out a priori. But in all cases, it is clear that questions of inference and induction are particularly central to machine learning and to computational biology.

When reasoning in the presence of certainty, one uses deduction. This is how the most advanced theories in information-poor sciences, such as physics or mathematics, are presented in an axiomatic fashion. Deduction is not controversial. The overwhelming majority of people agree on how to perform deductions in specific way: if X implies Y, and X is true, then Y must be true. This is the essence of Boole's algebra, and at the root of all our digital computers. When reasoning in the presence of uncertainty, one uses induction and inference: if X implies Y, and Y is true, then X is *more plausible*. An amazing and still poorly known fact is that there is a simple and unique consistent set of rules for induction, model selection, and comparison. It is called Bayesian inference. The Bayesian approach has been known for some time, but only recently has it started to infiltrate different areas of science and technology systematically, with useful results [229, 372, 373]. While machine learning may appear to some as an eclectic collection of models and learning algorithms, we believe the Bayesian framework provides a strong underlying foundation that unifies the different techniques. We now review the Bayesian

framework in general. In the following chapters, we apply it to specific classes of models and problems.

The Bayesian point of view has a simple intuitive description. The Bayesian approach assigns a degree of plausibility to any proposition, hypothesis, or model. (Throughout this book, hypothesis and model are essentially synonymous; models tend to be complex hypotheses with many parameters.) More precisely, in order properly to carry out the induction process, one ought to proceed in three steps:

1. Clearly state what the hypotheses or models are, along with *all* the background information and the data.

2. Use the language of probability theory to assign prior probabilities to the hypotheses.

3. Use probability calculus for the inference process, in particular to evaluate posterior probabilities (or degrees of belief) for the hypotheses in light of the available data, and to derive *unique* answers.

Such an approach certainly seems reasonable. Note that the Bayesian approach is not directly concerned with the creative process, how to generate new hypotheses or models. It is concerned only with assessing the value of models with respect to the available knowledge and data. This assessment procedure, however, may be very helpful in generating new ideas.

But why should the Bayesian approach be so compelling? Why use the language of probability theory, as opposed to any other method? The surprising answer to this question is that it can be proved, in a strict mathematical sense, that this is the only consistent way of reasoning in the presence of uncertainty. Specifically, there is a small set of very simple commonsense axioms, the Cox Jaynes axioms, under which it can be shown that the Bayesian approach is the unique consistent approach to inference and induction. Under the Cox Jaynes axioms, degrees of plausibility satisfy all the rules of probabilities exactly. Probability calculus is, then, all the machinery that is required for inference, model selection, and model comparison.

In the next section, we give a brief axiomatic presentation of the Bayesian point of view using the Cox Jaynes axioms. For brevity, we do not present any proofs or any historical background for the Bayesian approach, nor do we discuss any controversial issues regarding the foundations of statistics. All of these can be found in various books and articles, such as [51, 63, 122, 433, 284].

2.2 The Cox Jaynes Axioms

The objects we deal with in inference are propositions about the world. For instance, a typical proposition X is "Letter A appears in position i of sequence O." A proposition is either true or false, and we denote by $\bar{X}$ the complement of a proposition X. A hypothesis H about the world is a proposition, albeit a possibly complex one composed of the conjunction of many more elementary propositions. A model M can also be viewed as a hypothesis. The difference is that models tend to be very complex hypotheses involving a large number of parameters. In discussions where parameters are important, we will consider that $M = M(w)$, where w is the vector of all parameters. A complex model M can easily be reduced to a binary proposition in the form "Model M accounts for data D with an error level ϵ" (this vague statement will be made more precise in the following discussion). But for any purpose, in what follows there is no real distinction between models and hypotheses.

Whereas propositions are either true or false, we wish to reason in the presence of uncertainty. Therefore the next step is to consider that, given a certain amount of information I, we can associate with each hypothesis a degree of plausibility or confidence (also called degree or strength of belief). Let us represent it by the symbol $\pi(X|I)$. While $\pi(X|I)$ is just a symbol for now, it is clear that in order to have a scientific discourse, one should be able to compare degrees of confidence. That is, for any two propositions X and Y, either we believe in X more than in Y, or we believe in Y more than in X, or we believe in both equally. Let us use the symbol $>$ to denote this relationship, so that we write $\pi(X|I) > \pi(Y|I)$ if and only if X is more plausible than Y. It would be very hard not to agree that in order for things to be sensible, the relationship $>$ should be transitive. That is, if X is more plausible than Y, and Y is more plausible than Z, then X must be more plausible than Z. More formally, this is the first axiom,

$$\pi(X|I) > \pi(Y|I) \quad \text{and} \quad \pi(Y|I) > \pi(Z|I) \quad \text{imply} \quad \pi(X|I) > \pi(Z|I). \tag{2.1}$$

This axiom is trivial; it has, however, an important consequence: $>$ is an ordering relationship, and therefore degrees of belief can be expressed by real numbers. That is, from now on, $\pi(X|I)$ represents a number. This of course does not mean that such a number is easy to calculate, but merely that such a number exists, and the ordering among hypotheses is reflected in the ordering of real numbers. To proceed any further and stand a chance of calculating degrees of belief we need additional axioms or rules for relating numbers representing strengths of belief.

The amazing fact is that only two additional axioms are needed to constrain the theory entirely. This axiomatic presentation is usually attributed to

Cox and Jaynes [138, 283]. To better understand these two remaining axioms, the reader may imagine a world of very simple switches, where at each instant in time a given switch can be either on or off. Thus, all the elementary hypotheses or propositions in this world, at a given time, have the simple form "switch X is on" or "switch X is off." (For sequence analysis purposes, the reader may imagine that switch X is responsible for the presence or absence of the letter X, but this is irrelevant for a general understanding.) Clearly, the more confident we are that switch X is on (X), the less confident we are that switch X is off ($\bar{X}$). Thus, for any given proposition X, there should be a relationship between $\pi(X|I)$ and $\pi(\bar{X}|I)$. Without assuming anything about this relationship, it is sensible to consider that, all else equal, the relationship should be the same for all switches and for all types of background information, that is, for all propositions X and I. Thus, in mathematical terms, the second axiom states that there exists a function F such that

$$\pi(\bar{X}|I) = F[\pi(X|I)]. \tag{2.2}$$

The third axiom is only slightly more complicated. Consider this time two switches X and Y and the corresponding four possible joint states. Then our degree of belief that X is on and Y is off, for instance, naturally depends on our degree of belief that switch X is on, and our degree of belief that switch Y is off, *knowing that X is on*. Again, it is sensible that this relationship be independent of the switch considered and the type of background information I. Thus, in mathematical terms, the third axiom states that there exists a function G such that

$$\pi(X, Y|I) = G[\pi(X|I), \pi(Y|X, I)]. \tag{2.3}$$

So far, we have not said much about the information I. I is a proposition corresponding to the conjunction of all the available pieces of information. I can represent background knowledge, such as general structural or functional information about biological macromolecules. I can also include specific experimental or other data. When it is necessary to focus on a particular corpus of data D, we can write $I = (I, D)$. In any case, I is not necessarily fixed and can be augmented with, or replaced by, any number of symbols representing propositions, as already seen in the right-hand side of (2.3). When data are acquired sequentially, for instance, we may write $I = (I, D_1, \ldots, D_n)$. In a discussion where I is well defined and fixed, it can be dropped altogether from the equations.

The three axioms above entirely determine, up to scaling, how to calculate degrees of belief. In particular, one can prove that there is always a rescaling κ of degrees of belief such that $\mathbf{P}(X|I) = \kappa(\pi(X|I))$ is in $[0, 1]$. Furthermore, $\mathbf{P}$ is unique and satisfies all the rules of probability. Specifically, if degrees of belief are restricted to the $[0, 1]$ interval, then the functions F and G must be

given by $F(x) = 1 - x$ and $G(x, y) = xy$. The corresponding proof will not be given here and can be found in [138, 284]. As a result, the second axiom can be rewritten as the sum rule of probability,

$$\mathbf{P}(X|I) + \mathbf{P}(\bar{X}|I) = 1, \tag{2.4}$$

and the third axiom as the product rule,

$$\mathbf{P}(X, Y|I) = \mathbf{P}(X|I)\mathbf{P}(Y|X, I). \tag{2.5}$$

From here on, we can then replace degrees of confidence by probabilities. Note that if uncertainties are removed, that is, if $\mathbf{P}(X|I)$ is 0 or 1, then (2.4) and (2.5) yield, as a special case, the two basic rules of deduction or Boolean algebra, for the negation and conjunction of propositions [(1) "X or $\bar{X}$" is always true; (2) "X and Y" is true if and only if both X and Y are true]. By using the symmetry $\mathbf{P}(X, Y|I) = \mathbf{P}(Y, X|I)$ together with (2.5), one obtains the important Bayes theorem,

$$\mathbf{P}(X|Y, I) = \frac{\mathbf{P}(Y|X, I)\mathbf{P}(X|I)}{\mathbf{P}(Y|I)} = \mathbf{P}(X|I)\frac{\mathbf{P}(Y|X, I)}{\mathbf{P}(Y|I)}. \tag{2.6}$$

The Bayes theorem is fundamental because it allows inversion: interchanging conditioning and nonconditioning propositions. In a sense, it embodies inference or learning because it describes exactly how to update our degree of belief $\mathbf{P}(X|I)$ in X, in light of the new piece of information provided by Y, to obtain the new $\mathbf{P}(X|Y, I)$. $\mathbf{P}(X|I)$ is also called the prior probability, and $\mathbf{P}(X|Y, I)$, the posterior probability, with respect to Y. This rule can obviously be iterated as information becomes available. Throughout the book, $\mathbf{P}(X)$ is universally used to denote the probability of X. It should be clear, however, that the probability of X depends on the surrounding context and is not a universal concept. It is affected by the nature of the background information and by the space of alternative hypotheses under consideration.

Finally, one should be aware that there is a more general set of axioms for a more complete theory that encompasses Bayesian probability theory. These are the axioms of decision or utility theory, where the focus is on how to take "optimal" decisions in the presence of uncertainty [238, 63, 431] (see also appendix A). Not surprisingly, the simple axioms of decision theory lead one to construct and estimate Bayesian probabilities associated with the uncertain environment, and to maximize the corresponding expected utility. In fact, an even more general theory is game theory, where the uncertain environment includes other agents or players. Since the focus of the book is on data modeling only, these more general axiomatic theories will not be needed.

2.3 Bayesian Inference and Induction

We can now turn to the type of inference we are most interested in: deriving a parameterized model $M = M(w)$ from a corpus of data D. For simplicity, we will drop the background information I from the following equations. From Bayes theorem we immediately have

$$\mathbf{P}(M|D) = \frac{\mathbf{P}(D|M)\mathbf{P}(M)}{\mathbf{P}(D)} = \mathbf{P}(M)\frac{\mathbf{P}(D|M)}{\mathbf{P}(D)}. \tag{2.7}$$

The prior $\mathbf{P}(M)$ represents our estimate of the probability that model M is correct before we have obtained any data. The posterior $\mathbf{P}(M|D)$ represents our updated belief in the probability that model M is correct given that we have observed the data set D. The term $\mathbf{P}(D|M)$ is referred to as the likelihood.

For data obtained sequentially, one has

$$\mathbf{P}(M|D^1,\ldots,D^t) = \mathbf{P}(M|D^1,\ldots,D^{t-1})\frac{\mathbf{P}(D^t|M,D^1,\ldots,D^{t-1})}{\mathbf{P}(D^t|D^1,\ldots,D^{t-1})}. \tag{2.8}$$

In other words, the old posterior $\mathbf{P}(M|D^1,\ldots,D^{t-1})$ plays the role of the new prior. For technical reasons, probabilities can be very small. It is often easier to work with the corresponding logarithms, so that

$$\log\mathbf{P}(M|D) = \log\mathbf{P}(D|M) + \log\mathbf{P}(M) - \log\mathbf{P}(D). \tag{2.9}$$

To apply (2.9) to any class of models, we will need to specify the prior $\mathbf{P}(M)$ and the data likelihood $\mathbf{P}(D|M)$. Once the prior and data likelihood terms are made explicit, the initial modeling effort is complete. All that is left is cranking the engine of probability theory. But before we do that, let us briefly examine some of the issues behind priors and likelihoods in general.

2.3.1 Priors

The use of priors is a strength of the Bayesian approach, since it allows incorporating prior knowledge and constraints into the modeling process. It is sometimes also considered a weakness, on the ground that priors are subjective and different results can be derived with different priors. To these arguments, Bayesians can offer at least four different answers:

1. In general, the effects of priors diminish as the number of data increases. Formally, this is because the likelihood $-\log\mathbf{P}(D|M)$ typically increases linearly with the number of data points in D, while the prior $-\log\mathbf{P}(M)$ remains constant.

2. There are situations where objective criteria, such as maximum entropy and/or group invariance considerations, can be used to determine non-informative priors (for instance, [228]).

3. Even when priors are not mentioned explicitly, they are used implicitly. The Bayesian approach forces a clarification of one's assumption without sweeping the problem of priors under the rug.

4. Finally, and most important, the effects of different priors, as well as different models and model classes, can be assessed within the Bayesian framework by comparing the corresponding probabilities.

It is a matter of debate within the statistical community whether a general objective principle exists for the determination of priors in all situations, and whether maximum entropy (MaxEnt) is such a principle. It is our opinion that such a general principle does not really exist, as briefly discussed at the end of Appendix B. It is best to adopt a flexible and somewhat opportunistic attitude toward the selection of prior distributions, as long as the choices, as well as their quantitative consequences, are made explicit via the corresponding probabilistic calculations. MaxEnt, however, is useful in certain situations. For completeness, we now briefly review MaxEnt and group-theoretical considerations for priors, as well as three prior distributions widely used in practice.

Maximum Entropy

The MaxEnt principle states that the prior probability assignment should be the one with the maximum entropy consistent with all the prior knowledge or constraints (all information-theoretic notions, such as entropy and relative entropy, are reviewed for completeness in appendix B). Thus the resulting prior distribution is the one that "assumes the least," or is "maximally non-committal," or has the "maximum uncertainty." In the absence of any prior constraints, this leads of course to a uniform distribution corresponding to Laplace's "principle of indifference." Thus, when there is no information available on a parameter w, other than its range, a uniform prior over the range is a natural choice of prior. MaxEnt applies in modeling situations parametrized by a distribution P or by the corresponding histogram. MaxEnt is equivalent to using the entropic prior $\mathbf{P}(P) = e^{-\mathcal{H}(P)}/Z$, where $\mathcal{H}(P)$ is the entropy of P. MaxEnt is applied and further discussed in section 3.2. MaxEnt can also be viewed as a special case of an even more general principle, *minimum relative entropy* [486] (see appendix B).

Group-Theoretic Considerations

In many situations some, if not all, of the constraints on the prior distribution can be expressed in group-theoretical terms, such as invariance with respect to a group of transformations. A typical example is a scale parameter, such as the standard deviation σ of a Gaussian distribution. Suppose that we have only an idea of the range of σ, in the form $e^a < \sigma < e^b$. Then, within such range, the density $f(\sigma)$ of σ should be invariant to scaling of σ, and therefore f should be proportional to $d\sigma/\sigma$. By simple normalization, we find

$$f(\sigma) = \frac{1}{b-a}\frac{d\sigma}{\sigma}. \tag{2.10}$$

This is equivalent to having $\log\sigma$ uniformly distributed on the interval $[a, b]$ or having the densities of σ and σ^m identical. Other examples of group invariance analysis can be found in [282, 228].

Useful Practical Priors: Gaussian, Gamma, and Dirichlet

When prior distributions are not uniform, two useful and standard priors for continuous variables are the Gaussian (or normal) prior and the gamma prior. Gaussian priors with 0 mean are often used for the initialization of the weights between units in neural networks. A Gaussian prior, on a single parameter, has the form

$$\mathcal{N}(w|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(w-\mu)^2}{2\sigma^2}). \tag{2.11}$$

In the present context, one of the reasons the Gaussian distribution is preeminent is related to the maximum entropy principle. When the only information available about a continuous density is its mean μ and its variance σ^2, then the Gaussian density $\mathcal{N}(\mu,\sigma)$ is the one achieving maximal entropy [137] (see Appendix B).

The gamma density [177] with parameters α and λ is given by

$$\Gamma(w|\alpha,\lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)}w^{\alpha-1}e^{-\lambda w} \tag{2.12}$$

for $w > 0$, and 0 otherwise. $\Gamma(\alpha)$ is the gamma function $\Gamma(\alpha) = \int_0^\infty e^{-x}x^{\alpha-1}dx$. By varying α and λ and translating w, the gamma density allows a wide range of priors, with more mass concentrated in one specific region of parameter space. Gamma priors are useful whenever the range of a parameter is bounded on one side—for instance, in the case of a positive parameter such as a standard deviation ($\sigma \geq 0$).

Finally, in the case of multinomial distributions that play an essential role in this book, such as the choice of a letter from an alphabet at a given position

in a sequence, an important class of priors are the Dirichlet priors [63, 376]. By definition, a Dirichlet distribution on the probability vector $P = (p_1, \ldots, p_K)$, with parameters α and $Q = (q_1, \ldots, q_K)$, has the form

$$D_{\alpha Q}(P) = \frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha q_i)} \prod_{i=1}^{K} p_i^{\alpha q_i - 1} = \prod_{i=1}^{K} \frac{p_i^{\alpha q_i - 1}}{Z(i)}, \tag{2.13}$$

with $\alpha, p_i, q_i \geq 0$ and $\sum p_i = \sum q_i = 1$. For such a Dirichlet distribution, $\mathbf{E}(p_i) = q_i$, $\mathbf{Var}(p_i) = q_i(1 - q_i)/(\alpha + 1)$, and $\mathbf{Cov}(p_i p_j) = -q_i q_j/(\alpha + 1)$. Thus Q is the mean of the distribution, and α determines how peaked the distribution is around its mean. Dirichlet priors are important because they are the natural conjugate priors for multinomial distributions, as will be demonstrated in chapter 3. This simply means that the posterior parameter distribution, after having observed some data from a multinomial distribution with Dirichlet prior, also has the form of a Dirichlet distribution. The Dirichlet distribution can be seen as the multivariate generalization of the beta distribution, and can also be interpreted as a maximum entropy distribution over the space of distributions P, with a constraint on the average distance (i.e. relative entropy) to a reference distribution determined by Q and α (see appendix B).

2.3.2 Data Likelihood

In order to define $\mathbf{P}(D|M)$, one must come to grips with how a model M could also give rise to a different observation set D': in a Bayesian framework, sequence models *must* be probabilistic. A deterministic model assigns a probability 0 to all the data except the one it can produce exactly. This is clearly inadequate in biology and perhaps is one of the major lessons to be derived from the Bayesian point of view. Scientific discourse on sequence models—how well they fit the data and how they can be compared with each other—is *impossible* if the likelihood issue is not addressed honestly.

The likelihood question is clearly related to issues of variability and noise. Biological sequences are inherently "noisy," variability resulting in part from random events amplified by evolution. Mismatches and differences between specific individual sequences and the "average" sequence in a family, such as a protein family, are bound to occur and must be quantified. Because the same DNA or amino acid sequence will differ between individuals of the same species, and even more so across species, modelers always need to think in probabilistic terms. Indeed, a number of models used in the past in a more or less heuristic way, without clear reference to probabilities, are suddenly illuminated when the probabilistic aspects are made explicit. Dealing with

the probabilistic aspects not only clarifies the issues and allows a rigorous discourse, but often also suggests new modeling avenues.

The computation of the likelihood is of course model-dependent and cannot be addressed in its generality. In section 2.4, we will outline some general principles for the derivation of models where the likelihood can be estimated without too many difficulties. But the reader should be aware that whatever criterion is used to measure the difference or error between a model and the data, such a criterion always comes with an underlying probabilistic model that needs to be clarified and is amenable to Bayesian analysis. Indeed, if the fit of a model $M = M(w)$ with parameters w is measured by some error function $f(w, D) \geq 0$ to be minimized, one can always define the associated likelihood to be

$$\mathbf{P}(D|M(w)) = \frac{e^{-f(w,D)}}{Z}, \tag{2.14}$$

where $Z = \int_w e^{-f(w,D)} dw$ is a normalizing factor (the "partition function" in statistical mechanics) that ensures the probabilities integrate to 1. As a result, minimizing the error function is equivalent to maximum likelihood (ML) estimation, or more generally maximum a posteriori (MAP) estimation. In particular, when the sum of squared differences is used to compare quantities, a rather common practice, this implies an underlying Gaussian model. Thus the Bayesian point of view clarifies the probabilistic assumptions that *must* underlie any criteria for matching models with data.

2.3.3 Parameter Estimation and Model Selection

We now return to the general Bayesian inference machinery. Two specific models M_1 and M_2 can be compared by comparing their probabilities $\mathbf{P}(M_1|D)$ and $\mathbf{P}(M_2|D)$. One objective often is to find or approximate the "best" model within a class—that is, to find the set of parameters w maximizing the posterior $\mathbf{P}(M|D)$, or $\log \mathbf{P}(M|D)$, and the corresponding error bars (see appendix A). This is called MAP estimation. In order to deal with positive quantities, this is also equivalent to minimizing $-\log \mathbf{P}(M|D)$:

$$\mathcal{E} = -\log \mathbf{P}(M|D) = -\log \mathbf{P}(D|M) - \log \mathbf{P}(M) + \log \mathbf{P}(D). \tag{2.15}$$

From an optimization standpoint, the logarithm of the prior plays the role of a regularizer, that is, of an additional penalty term that can be used to enforce additional constraints, such as smoothness. Note that the term $\mathbf{P}(D)$ in (2.15) plays the role of a normalizing constant that does not depend on the parameters w, and is therefore irrelevant for this optimization. If the prior $\mathbf{P}(M)$ is uniform over all the models considered, then the problem reduces to finding the maximum of $\mathbf{P}(D|M)$, or $\log \mathbf{P}(D|M)$. This is just ML estimation. In

summary, a substantial portion of this book and of machine-learning practice is based on MAP estimation, that is, the minimization of

$$\mathcal{E} = -\log \mathbf{P}(D|M) - \log \mathbf{P}(M), \tag{2.16}$$

or even the simpler ML estimation, that is, the minimization of

$$\mathcal{E} = -\log \mathbf{P}(D|M). \tag{2.17}$$

In most interesting models, the function being optimized is complex and its modes cannot be solved analytically. Thus one must resort to iterative and possibly stochastic methods such as gradient descent or simulated annealing, and also settle for approximate or suboptimal solutions.

Bayesian inference, however, is iterative. Finding a highly probable model within a certain class is only its first level. Whereas finding the optimal model is common practice, it is essential to note that this is really useful only if the distribution $\mathbf{P}(M|D)$ is sharply peaked around a unique optimum. In situations characterized by a high degree of uncertainty and relatively small amounts of data available, this is often not the case. Thus a Bayesian is really interested in the function $\mathbf{P}(M|D)$ over the entire space of models rather than in its maxima only, and more precisely in evaluating expectations with respect to $\mathbf{P}(M|D)$. This leads to higher levels of Bayesian inference, as in the case of prediction problems, marginalization of nuisance parameters, and class comparisons.

2.3.4 Prediction, Marginalization of Nuisance Parameters, and Class Comparison

Consider a prediction problem in which we are trying to predict the output value y of an unknown parameterized function f_w, given an input x. It is easy to show that the optimal prediction is given by the expectation

$$\mathbf{E}(y) = \int_w f_w(x)\mathbf{P}(w|D)dw. \tag{2.18}$$

This integral is the average of the predictions made by each possible model f_w, weighted by the plausibility $\mathbf{P}(w|D)$ of each model. Another example is the process of marginalization, where integration of the posterior parameter distribution is carried out only with respect to a subset of the parameters, the so-called nuisance parameters [225]. In a frequentist framework, where probabilities are defined in terms of observed frequencies, the notion of distribution over the parameters is not defined, and therefore nuisance parameters cannot be integrated out easily. Finally, one is often led to the problem of comparing two model classes, C_1 and C_2. To compare C_1 and C_2 in the Bayesian

framework, one must compute $\mathbf{P}(C_1|D)$ and $\mathbf{P}(C_2|D)$ using Bayes' theorem: $\mathbf{P}(C|D) = \mathbf{P}(D|C)\mathbf{P}(C)/\mathbf{P}(D)$. In addition to the prior $\mathbf{P}(C)$, one must calculate the *evidence* $\mathbf{P}(D|C)$ by averaging over the entire model class:

$$\mathbf{P}(D|C) = \int_{w \in C} \mathbf{P}(D, w|C)dw = \int_{w \in C} \mathbf{P}(D|w, C)\mathbf{P}(w|C)dw. \tag{2.19}$$

Similar integrals also arise with hierarchical models and hyperparameters (see below). In cases where the likelihood $\mathbf{P}(D|w, C)$ is very peaked around its maximum, such expectations can be approximated using the mode, that is, the value with the highest probability. But in general, integrals such as (2.18) and (2.19) require better approximations—for instance using Monte Carlo sampling methods [491, 396, 69], as briefly reviewed in chapter 4. Such methods, however, are computationally intensive and not always applicable to the models to be considered. This book is mostly concerned only with likelihood calculations and the first level of Bayesian inference (ML and MAP). The development of methods for handling higher levels of inference is an active area of research, and these should be considered whenever possible. The available computer power is of course an important issue in this context.

2.3.5 Ockham's Razor

As a final point raised in section 2.1, it does not make sense to choose a simple model on the basis that available data are scarce. *Everything else being equal*, however, it is true that one should prefer a simple hypothesis to a complex one. This is Ockham's razor. As pointed out by several authors, Ockham's razor is automatically embodied in the Bayesian framework [285, 373] in at least two different ways. In the first, trivial way, one can introduce priors that penalize complex models. But even without such priors, parameterized complex models will tend to be consistent with a larger volume of data space. Since a likelihood $\mathbf{P}(D|M)$ must sum to 1 over the space of data, if $\mathbf{P}(D|M)$ covers a larger expanse of data space, the likelihood values for given data sets will be smaller on average. Therefore, all else equal, complex models will tend to assign a correspondingly smaller likelihood to the observed data.

2.3.6 Minimum Description Length

An alternative approach to modeling is the *minimum description length* (MDL) [446]. MDL is related to ideas of data compression and information transmission. The goal is to transmit the data over a communication channel. Transmitting the data "as is" is not economical: nonrandom data contains structure and redundancies, and therefore must be amenable to compression. A good

model of the data should capture their structure and yield good compression. The optimal model is the one that minimizes the length of the total message required to describe the data. This includes both the length required to specify the model itself and the data given the model. To a first approximation, MDL is closely related to the Bayesian point of view. According to Shannon's theory of communication [483], the length of the message required to communicate an event that has probability p is proportional to $-\log p$. Thus the most probable model has the shortest description. Some subtle differences between MDL and the Bayesian point of view can exist, however, but these will not concern us here.

2.4 Model Structures: Graphical Models and Other Tricks

Clearly, the construction or selection of suitable models is dictated by the data set, as well as by the modeler's experience and ingenuity. It is, however, possible to highlight a small number of very general techniques or tricks that can be used to shape the structure of the models. Most models in the literature can be described in terms of combinations of these simple techniques. Since in machine learning the starting point of any Bayesian analysis is almost always a high-dimensional probability distribution $\mathbf{P}(M, D)$ and the related conditional and marginal distributions (the posterior $\mathbf{P}(M|D)$, the likelihood $\mathbf{P}(D|M)$, the prior $\mathbf{P}(M)$, and the evidence $\mathbf{P}(D)$); these rules can be seen as ways of decomposing, simplifying, and parameterizing such high-dimensional distributions.

2.4.1 Graphical Models and Independence

By far the most common simplifying trick is to assume some independence between the variables or, more precisely, some conditional independence of subsets of variables, conditioned on other subsets of variables. These independence relationships can often be represented by a graph where the variables are associated with the nodes, and a missing edge represents a particular independence relationship (precise definitions can be found in appendix C). See [416, 350, 557, 121, 499, 106, 348, 286] for general reviews, treatments, or pointers to the large literature on this topic.

The independence relationships result in the fundamental fact that the global high-dimensional probability distribution, over all variables, can be factored into a product of simpler local probability distributions over lower-dimensional spaces associated with smaller clusters of variables. The clusters are reflected in the structure of the graph.

Graphical models can be subdivided into two broad categories depending on whether the edges of the associated graph are directed or undirected. Undi-

rected edges are typical in problems where interactions are considered to be symmetric, such as in statistical mechanics or image processing [272, 199, 392]. In the undirected case, in one form or another, these models are called Markov random fields, undirected probabilistic independence networks, Boltzmann machines, Markov networks, and log-linear models.

Directed models are used in cases where interactions are not symmetric and reflect causal relationships or time irreversibility [416, 286, 246]. This is typically the case in expert systems and in all problems based on temporal data. The Kalman filter, a tool widely used in signal processing and control, can be viewed in this framework. In the case of temporal series, the independence assumptions are often those used in Markov models. Not surprisingly, most if not all of the models discussed in this book—NNs and HMMs in particular—are examples of graphical models with directed edges. A systematic treatment of graphical models in bioinformatics is given in chapter 9. Typical names for such models in the literature are Bayesian networks, belief networks, directed probabilistic independence networks, causal networks, and influence diagrams. It is also possible to develop a theory for the mixed case [557], where both directed and undirected edges are present. Such mixed graphs are also called chain independence graphs. The basic theory of graphical models is reviewed in appendix C.

Here we introduce the notation needed in the following chapters. By $G = (V, E)$ we denote a graph G with a set V of vertices and a set E of edges. If the edges are directed, we write $G = (V, \vec{E})$. In an undirected graph, $N(i)$ represents the sets of all the neighbors of vertex i, and $C(i)$ represents the set of all the vertices that are connected to i by a path. So,

$$N(i) = \{j \in V : (i, j) \in E\}. \tag{2.20}$$

In a directed graph, we use the obvious notation $N^-(i)$ and $N^+(i)$ to denote all the parents of i and all the children of i, respectively. Likewise, $C^-(i)$ and $C^+(i)$ denote the ancestors, or the "past," and the descendants of i, or the "future" of i. All these notations are extended in the obvious way to any set of vertices I. So for any $I \subseteq V$,

$$N(I) = \{j \in V : \exists i \in I \quad (i, j) \in E\} - I. \tag{2.21}$$

This is also called the boundary of I.

One fundamental observation is that in most applications the resulting graphs are *sparse*. Thus the global probability distribution factors into a relatively small number of relatively small local distributions. And this is key to the implementation of efficient computational structures for learning and inference, based on the local propagation of information between clusters of variables in the graph. The following techniques are not independent of the general graphical model ideas, but can often be viewed as special cases.

2.4.2 Hidden Variables

In many models, it is typical to assume that the data result in part from the action of hidden or latent variables, or causes, that either are not available in the data gathered or perhaps are fundamentally unobservable [172]. Missing data can also be treated as a hidden variable. The activations of the hidden units of a network, or the state sequence of an HMM, are typical examples of hidden variables. Another example is provided by the coefficients of a mixture (see below). Obviously the parameters of a model, such as the weights of an NN or the emission/transition probabilities of an HMM, could also be regarded as hidden variables in some sense, although this would be an unconventional terminology. Typical inference problems in hidden variable models are the estimation of the probability distribution over the set of hidden variables, its modes, and the corresponding expectations. These often appear as subproblems of the general parameter estimation problem in large parameterized models, such as HMMs. An important algorithm for parameter estimation with missing data or hidden variable is the EM algorithm, described in chapter 4 and further demonstrated in chapter 7 on HMMs.

2.4.3 Hierarchical Modeling

Many problems have a natural hierarchical structure or decomposition. This can result, for instance, from the existence of different time scales or length scales in the problem. The clusters described above in the general section on graphical modeling can also be viewed as nodes of a higher-level graphical model for the data (see, for instance, the notion of junction tree in [350]). In a related but complementary direction, the prior on the parameters of a model can have a hierarchical structure in which parameters at one level of the hierarchy are used to define the prior distribution on the parameters at the next level in a recursive way, with the number of parameters typically decreasing at each level as one ascends the hierarchy. All the parameters above a given level are often called "hyperparameters" with respect to that level.

Hyperparameters are used to provide more flexibility while keeping control over the complexity and structure of the model. Hyperparameters have "high gain" in the sense that small hyperparameter variations can induce large model variations at the level below. Hyperparameters also allow for parameter reduction because the model prior can be calculated from a (usually smaller) number of hyperparameters. In symbolic form,

$$\mathbf{P}(w) = \int_{\alpha} \mathbf{P}(w|\alpha)\mathbf{P}(\alpha)d\alpha, \tag{2.22}$$

where α represents hyperparameters for the parameter w with prior $\mathbf{P}(\alpha)$. As a typical example, consider the connection weights in a neural network. In a given problem, it may be a good idea to model the prior on a weight by using a Gaussian distribution with mean μ and standard deviation σ. Having a different set of hyperparameters μ and σ for each weight may yield a model with too few constraints. All the σs of a given unit, or in an entire layer, can be tied and assumed to be identical. At a higher level, a prior can be defined on the σs, and so on. An example of a hierarchical Dirichlet model is given in appendix D.

2.4.4 Hybrid Modeling/Parameterization

Parameterization issues are important in machine learning, if only because the models used are often quite large. Even when the global probability distribution over the data and the parameters has been factored into a product of simpler distributions, as a result of independence assumptions, one often must still parameterize the component distributions. Two useful general approaches for parameterizing distributions are mixture models and neural networks.

In mixture models, a complex distribution P is parameterized as a linear convex combination of simpler or canonical distributions in the form

$$P = \sum_{i=1}^{n} \lambda_i P_i, \tag{2.23}$$

where the $\lambda_i \geq 0$ are called the mixture coefficients and satisfy $\sum_i \lambda_i = 1$. The distributions P_i are called the components of the mixture and can carry their own parameters (means, standard deviations, etc.). A review of mixture models can be found in [173, 522].

Neural networks are also used to reparameterize models, that is, to compute model parameters as a function of inputs and connection weights. As we shall see, this is in part because neural networks have universal approximation properties and good flexibility, combined with simple learning algorithms. The simplest example is perhaps in regression problems, where a neural network can be used to calculate the mean of the dependent variable as a function of the independent variable, the input. A more subtle example will be given in chapter 9, where neural networks are used to calculate the transition and emission parameters of an HMM. The term "hybrid" is sometimes used to describe situations in which different model classes are combined, although the combination can take different forms.

2.4.5 Exponential Family of Distributions

The exponential family of distributions is briefly reviewed in appendix A. Here it suffices to say that many of the most commonly used distributions (Gaussian, multinomial, etc.) belong to this family, and that using members of the family often leads to computationally efficient algorithms. For a review of the exponential family, with a comprehensive list of references, see [94].

2.5 Summary

We have briefly presented the Bayesian approach to modeling and inference. The main advantage of a Bayesian approach is obvious: it provides a principled and rigorous approach to inference, with a strong foundation in probability theory. In fact, one of the most compelling reasons in favor of Bayesian induction is its uniqueness under a very small set of commonsense axioms. We grant that mathematicians may be more receptive than biologists to such an argument.

The Bayesian framework clarifies a number of issues, on at least three different levels. First, a Bayesian approach forces one to clarify the prior knowledge, the data, and the hypotheses. The Bayesian framework is entirely open to, and actually encourages, questioning any piece of information. It deals with the subjectivity inherent in the modeling process not by sweeping it under the rug but, rather, by incorporating it up front in the modeling process. It is fundamentally an iterative process where models are progressively refined. Second, and this is perhaps the main lesson here, sequence models *must* be probabilistic and come to grips with issues of noise and variability in the data, in a quantifiable way. Without this step it is impossible to have a rigorous scientific discourse on models, to determine how well they fit the data, and ultimately to compare models and hypotheses. Third, the Bayesian approach clarifies how to proceed with inference, that is, how to compare models and quantify errors and uncertainties, basically by cranking the engine of probability. In particular, it provides unambiguous, unique answers to well-posed questions. It defines the set of rules required to play a fair modeling game. The basic step is to compute model plausibilities, with respect to the available data and the associated expectations, using the rules of probability theory and possibly numerical approximations.

The Bayesian approach can lead to a better understanding of the weaknesses of a model, and thereby help in generating better models. In addition, an objective way of comparing models, and of making predictions based on models, will become more important as the number, scope, and complexity of models for biological macromolecules, structure, function, and regulation in-

crease. Issues of model comparison and prediction will become progressively more central as databases grow in size and complexity. New ideas are likely to emerge from the systematic application of Bayesian probabilistic ideas to sequence analysis problems.

The main drawback of the Bayesian approach is that it can be computationally intensive, especially when averages need to be computed over high-dimensional distributions. For the largest sequence models used in this book, one is unlikely to be able to carry out a complete Bayesian integration on currently available computers. But continuing progress in Monte Carlo [491, 69] and other approximation techniques, as well as steady increases in raw computing power in workstations and parallel computers, is encouraging.

Once the general probabilistic framework is established, the next central idea is that of graphical models: to factor high-dimensional probability distributions by exploiting independence assumptions that have a graphical substrate. Most machine-learning models and problems can be represented in terms of recursive sparse graphs, at the levels of both the variables involved, observed or hidden, and the parameters. Sparse recursive graphs appear as a universal underlying language or representational structure for most models and machine-learning applications.

This page intentionally left blank