Studies in Computational Intelligence 286

Alexander Bolshoy
Zeev (Vladimir) Volkovich
Valery Kirzhner
Zeev Barzily

Genome Clustering

From Linguistic Models to Classification of Genetic Texts

Springer

Alexander Bolshoy, Zeev (Vladimir) Volkovich, Valery Kirzhner, and Zeev Barzily

Genome Clustering

Studies in Computational Intelligence, Volume 286

Further volumes of this series can be found on our homepage: springer.com

Vol. 265. Zbigniew W. Ras and Li-Shiang Tsay (Eds.)
Advances in Intelligent Information Systems, 2009
ISBN 978-3-642-05182-1

Vol. 266. Akitoshi Hanazawa, Tsutom Miki, and Keiichi Horio (Eds.)
Brain-Inspired Information Technology, 2009
ISBN 978-3-642-04024-5

Vol. 267. Ivan Zelinka, Sergej Celikovský, Hendrik Richter, and Guanrong Chen (Eds.)
Evolutionary Algorithms and Chaotic Systems, 2009
ISBN 978-3-642-10706-1

Vol. 268. Johann M.Ph. Schumann and Yan Liu (Eds.)
Applications of Neural Networks in High Assurance Systems, 2009
ISBN 978-3-642-10689-7

Vol. 269. Francisco Fernández de de Vega and Erick Cantú-Paz (Eds.)
Parallel and Distributed Computational Intelligence, 2009
ISBN 978-3-642-10674-3

Vol. 270. Zong Woo Geem
Recent Advances In Harmony Search Algorithm, 2009
ISBN 978-3-642-04316-1

Vol. 271. Janusz Kacprzyk, Frederick E. Petry, and Adnan Yazici (Eds.)
Uncertainty Approaches for Spatial Data Modeling and Processing, 2009
ISBN 978-3-642-10662-0

Vol. 272. Carlos A. Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan (Eds.)
Advances in Multi-Objective Nature Inspired Computing, 2009
ISBN 978-3-642-11217-1

Vol. 273. Fatos Xhafa, Santi Caballé, Ajith Abraham, Thanasis Daradoumis, and Angel Alejandro Juan Perez (Eds.)
Computational Intelligence for Technology Enhanced Learning, 2010
ISBN 978-3-642-11223-2

Vol. 274. Zbigniew W. Raś and Alicja Wieczorkowska (Eds.)
Advances in Music Information Retrieval, 2010
ISBN 978-3-642-11673-5

Vol. 275. Dilip Kumar Pratihar and Lakhmi C. Jain (Eds.)
Intelligent Autonomous Systems, 2010
ISBN 978-3-642-11675-9

Vol. 276. Jacek Mańdziuk
Knowledge-Free and Learning-Based Methods in Intelligent Game Playing, 2010
ISBN 978-3-642-11677-3

Vol. 277. Filippo Spagnolo and Benedetto Di Paola (Eds.)
European and Chinese Cognitive Styles and their Impact on Teaching Mathematics, 2010
ISBN 978-3-642-11679-7

Vol. 278. Radomir S. Stankovic and Jaakko Astola
From Boolean Logic to Switching Circuits and Automata, 2010
ISBN 978-3-642-11681-0

Vol. 279. Manolis Wallace, Ioannis E. Anagnostopoulos, Phivos Mylonas, and Maria Bielikova (Eds.)
Semantics in Adaptive and Personalized Services, 2010
ISBN 978-3-642-11683-4

Vol. 280. Chang Wen Chen, Zhu Li, and Shiguo Lian (Eds.)
Intelligent Multimedia Communication: Techniques and Applications, 2010
ISBN 978-3-642-11685-8

Vol. 281. Robert Babuska and Frans C.A. Groen (Eds.)
Interactive Collaborative Information Systems, 2010
ISBN 978-3-642-11687-2

Vol. 282. Husrev Taha Sencar, Sergio Velastin, Nikolaos Nikolaidis, and Shiguo Lian (Eds.)
Intelligent Multimedia Analysis for Security Applications, 2010
ISBN 978-3-642-11754-1

Vol. 283. Ngoc Thanh Nguyen, Radoslaw Katarzyniak, and Shyi-Ming Chen (Eds.)
Advances in Intelligent Information and Database Systems, 2010
ISBN 978-3-642-12089-3

Vol. 284. Juan R. González, David Alejandro Pelta, Carlos Cruz, Germán Terrazas, and Natalio Krasnogor (Eds.)
Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), 2010
ISBN 978-3-642-12537-9

Vol. 285. Roberto Cipolla, Sebastiano Battiato, and Giovanni Maria Farinella (Eds.)
Computer Vision, 2010
ISBN 978-3-642-12847-9

Vol. 286. Alexander Bolshoy, Zeev (Vladimir) Volkovich, Valery Kirzhner, and Zeev Barzily
Genome Clustering, 2010
ISBN 978-3-642-12951-3

Alexander Bolshoy, Zeev (Vladimir) Volkovich,
Valery Kirzhner, and Zeev Barzily

Genome Clustering

From Linguistic Models to Classification of Genetic Texts

Springer

Alexander Bolshoy
The Department of Evolutionary and
Environmental Biology and
the Institute of Evolution
University of Haifa, Haifa 39105
Israel
E-mail: bolshoy@research.haifa.ac.il

Valery Kirzhner
The Institute of Evolution
University of Haifa,
Haifa 39105
Israel
E-mail: valery@research.haifa.ac.il

Zeev (Vladimir) Volkovich
Software Engineering Department
ORT Braude College
P.O. Box: 78
Karmiel, 20101
Israel
E-mail: vlvolkov@ort.org.il

Zeev Barzily
Software Engineering Department
ORT Braude College
P.O. Box: 78
Karmiel, 20101
Israel
E-mail: zbarzily@ort.org.il

ISBN 978-3-642-12951-3 e-ISBN 978-3-642-12952-0

DOI 10.1007/978-3-642-12952-0

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2010926032

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Foreword

Knighting in sequence biology

Edward N. Trifonov

Genome classification, construction of phylogenetic trees, became today a major approach in studying evolutionary relatedness of various species in their vast diversity. Although the modern genome clustering delivers the trees which are very similar to those generated by classical means, and basic terminology is the same, the phenotypic traits and habitats are not anymore the playground for the classification. The sequence space is the playground now. The phenotypic traits are replaced by sequence characteristics, "words", in particular. Matter-of-factually, the phenotype and genotype merged, to confusion of both classical and modern phylogeneticists.

Accordingly, a completely new vocabulary of stringology, information theory and applied mathematics took over. And a new brand of scientists emerged – those who do know the math and, simultaneously, (do?) know biology.

The book is written by the authors of this new brand. There is no way to test their literacy in biology, as no biologist by training would even try to enter into the elite circle of those who masters their almost occult language. But the army of informaticians, formal linguists, mathematicians humbly (or aggressively) longing to join modern biology, got an excellent introduction to the field of genome clustering, written by the team of their kin.

The analogy genomic sequences – texts is both an immediate simple thought, and an open door to the depths of genetic information and intricacies of its organization. The most fascinating and unique features of these texts are multiplicity, degeneracy and overlapping of various codes carried by the genetic sequences. In this respect mere transfer of techniques used for analysis of familiar "monocode" texts to the "polycode" sequences would be naïve. But no one would deny importance of such transfer, to begin with, to reveal, at least, the amazing specifics of the new reality. Another interesting aspect of the genomes is the uncertainty of the species' formal definition. Already in classical genetics this was a stumbling block. The fertile progeny based definition of Dobzhansky[1], though broadly accepted, does not fit all diversity of species. In the genomics the matter becomes even more complicated, in particular, due to horizontal gene transfer. It appears that the species is not an elementary node of evolution. Rather, the gene, or (again uncertain) DNA segment in general, is the node.

Principally new techniques have to be introduced to cope with this very special language. The monograph is a rather comprehensive outline of the state of art in the field, introducing as well some original developments. The appreciation of the principal differences of the natural sequence language from all we knew before is an important merit of the book.

1. Dobzhansky, T.: Genetics and the Origin of Species. Columbia Univ. Press, New York (1937)

Preface

People like to compare and do it in a great variety of fields and with all kinds of objects. In particular, comparative biological studies of different species of living beings lead us to better understanding of known biological phenomena and even to novel discoveries in the biological science. In modern biology, species are often presented by their genomes. Thus, instead of comparing external organisms' features such as the length of the tail, the shape of the wings, etc., it is possible now to compare organisms' genomes, which are represented as long texts over the alphabet of four letters.

There exist different methods of analyzing texts which are written in human languages or composed of special symbols (e.g., computer programs). Although these methods had been developed long before the discovery of genetic texts, many of them are applicable to the genomic text analysis as well, and some are described in this book. However, there also exist methods which were not borrowed from the studies of natural languages, but were developed especially for the comparison of genomic texts.

This book deals with the methods of text comparison which are based on different techniques of converting the text into a distribution on a certain finite support, be it a genetic text or a text of some other type. Such distribution is usually referred to as "spectrum". The measure of dissimilarity of two texts is formally expressed as a certain "distance" between the spectra of these texts. Such definition implies that the similarity of the texts results from the similarity of the random processes generating the texts. It is obvious, thus, that the zero distance between two texts does not necessarily imply their letter-by-letter coincidence; for example, the texts may be just different implementations of the same process. The spectrum support usually represents a finite set of words. In a natural language, the latter may be the words of the language, while in a genetic text, particular patterns may be considered as words. The patterns range from the simplest signals to genes, which are parts of the genetic text. However, in the natural language analysis, formal, meaningless words, which are called *N*-grams, are also successfully employed. Since the repertoire of different patterns in genetic texts is relatively small, the use of *N*-grams for the genetic text analysis appears to be still more beneficial. In certain applications, the spectrum support may be a set of relative positions in the text, but in this case, too, the distribution value in each position is evaluated as some function of words which are connected, in a certain way, with each position. The fact that these are the words, whatever their definition may be, that are used as the basis for the spectrum evaluation, allows viewing the methods under consideration as a part of a more general field which may be called “DNA linguistics”.

Genetic texts have certain features which are used for their analysis. The essential features, as well as some relevant information on the molecular biology of the cell, are presented in Chapter 1. Additionally, the reader can refer to several excellent introductory courses such as [8], [297] and [184].

Since this book is dedicated to the methods of genetic sequence comparison or, in other words, to a particular approach to genetic text classification, we review some classical general approaches to classification in Chapter 2. This chapter provides a brief introduction to the Linnaean classification system, to modern *taxonomy*, and to the field of molecular evolution called *phylogenetic systematics*. In the text of this book, we often compare the described results with the above classifications. The following books may be recommended for further reading on the topic of molecular evolution: [95], [204] and [200].

Chapter 3 provides a review of the main *data mining* models generating the text spectra which were developed for the analysis of texts written in natural languages. In particular, in the framework of some models, the coincidence (or similarity) of the spectra suggests the common author or the same topic of the two documents. The models which are based on the “letter-by-letter” comparison of texts are also described. They are further used in the book for constructing the spectra of genetic texts.

In Chapter 4, the questions are discussed as to the standpoint from which the DNA molecule can be viewed as a certain text and how this text can be evaluated in terms of formal grammar. Another essential question considered in the chapter concerns the process of creating genetic texts. While texts in natural languages are written by people, countless numbers of genetic texts (a unique text for each species and even, as it appears now, a unique text for each individual organism’s genome) are “written” in the course of evolution. The models of special mechanisms which evolution uses for writing genetic texts are also described in the chapter. Obviously, the fact that *DNA texts* are, actually, the result of the evolution process should be employed for the comparison of these texts.

In Chapter 5, the particular case of *digrams* (*N*-gram with *N*=2) is described in detail, including the results of bacterial genome classification obtained by this method. Moreover, the concepts of *fuzzy N-grams* and of compositional spectra (CS) based on such *N*-grams are introduced. The evaluation of CS is a complicated computational problem; hence some plausible algorithms for its solution are also discussed in the chapter. Quite a few examples of genetic texts are employed to assess the properties and the biological appropriateness of different distance functions.

Chapter 6 elaborates on the *application of the CS* model to the genome classification; in particular, the optimal parameters of the model are obtained. Finally, two possible classifications of species “across life” are derived and their relevance to the standard classification is discussed.

In Chapter 7, a different *profile-based approach* to classification is presented. As a result of the suggested technique, the text is converted to a point in the

K-dimensional Euclidean space. The general description of the profile-construction method is followed by consideration of two important applications: in the first example, the *linguistic complexity* measure is employed, while in the second example, the measure based on *DNA curvature* is used.

In Chapter 8, the new approach to phylogenetics based on considering the whole-genome information is illustrated. This approach, called *phylogenomics*, is closely related to the main topic of the book since it also deals with embedding of the genome into a coordinate space. The sets of all the genes of particular prokaryotic genomes were used in the framework of the Information Bottleneck method adapted for genome clustering. The dendrogram of the genome classification obtained by this method represents, actually, a phylogenetic tree.

In Appendix A, the reader is introduced to the main ideas and techniques of the *clustering* approach to classification.

In Appendix B, a short review of three *sequence complexity* measure methods is compiled.

Appendix C is devoted to the introduction to the issue of *DNA curvature*.

The book is written by four co-authors, whose fields of expertise are close, but still represent different lines of research. Therefore, it would be virtually impossible to bring in harmony a great many details and maintain a uniform style of the text without the help of our persistent and careful scientific editor, Tanya Pyatigorskaya, PhD in molecular biophysics, to whom the authors express their deep gratitude.

Contents

Chapter 1
Biological Background

This chapter is intended as a brief introduction to those fields of molecular biology which are essential for understanding the content of the book[1]. We give here a general idea about the molecular basis of heredity, in other words, the molecular mechanisms of storing and implementing information about an organism's traits and of transmitting genetic information to the next generation. In this way, we also lay the foundation for those molecular mechanisms of evolution that are relevant to the methods described in this book. To achieve the above goals, we introduce the reader to the spectacular process of the genetic information flow from DNA molecules. This process directs the whole activity of the living cell, including duplication of genetic material and protein synthesis.

1.1 The Cell

The cell is the smallest structural unit of an organism which is capable of independent functioning. Its components are a *nucleus*, *cytoplasm*[2], various organelles, and a *membrane*, which surrounds the cell. There exist a lot of various types of cells, but the main division is into *prokaryotes* and *eukaryotes*. A prokaryotic cell usually represents a unicellular organism and thus has to carry out all the organism's functions. Some eukaryotic cells may also represent unicellular organisms such as the famous *Amoeba proteus*, but most of them constitute multicellular organisms such as plants, animals, fungi, humans and are differentiated with respect to their functions.

1.1.1 Prokaryotic Cell

The essential, defining feature of a prokaryotic cell is the lack of a membrane-bound cell nucleus. The genetic material of prokaryotes is usually concentrated in a single circular double-stranded DNA molecule; besides, some

[1] Those readers who have taken an elementary course in molecular biology can turn directly to the next chapter.

[2] **Cytoplasm** is a gelatinous, semi-transparent fluid that fills a cell.

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 1–16, 2010.
springerlink.com

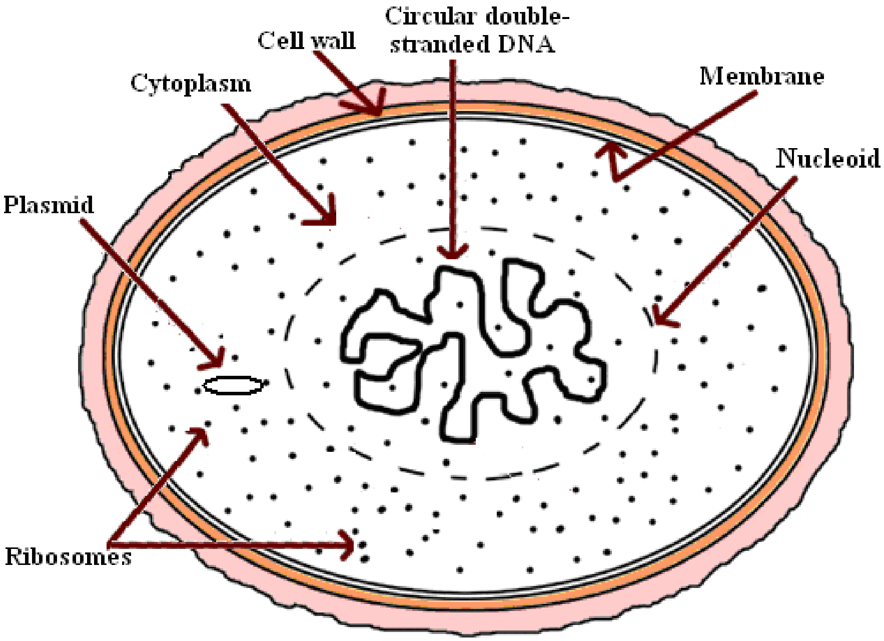


Fig. 1.1 Prokaryotic cell

genes are stored on one or more plasmids (see section 1.5). Prokaryotic cells (Fig. 1.1) are relatively small (usually 1 - 10 μm in diameter), being, however, larger than viruses (the size of about 0.1 μm). The internal structure of prokaryotes is much simpler than that of eukaryotes in that the former lack membrane-bound organelles except for ribosomes (see subsection 1.4.2.2). Prokaryotes are traditionally classified (see Chapter 2) within two kingdoms: Eubacteria (such as the well-known *Escherichia coli*, found, in particular, in the human intestine) and *Archaebacteria* (initially, thought to live in extreme environments), which differ biochemically and genetically in a number of features.

1.1.2 Eukaryotic Cell

The characteristic feature of a eukaryotic cell which distinguishes it from a prokaryotic one is the presence of a nucleus - a spherical body, surrounded by a thin membrane, that contains the organism's genetic material, structurally organized in chromosomes (see Section 1.5). Typical eukaryotic cells have linear dimensions (100 μm – 1 mm) much larger than those of prokaryotes and, which is most important, the former are much more structurally and functionally complex. In particular, the cytoplasm surrounding the nucleus of a eukaryotic cell contains membrane-bound subunits, called organelles (see Fig. 1.2), which carry out specialized reactions within their boundaries and,

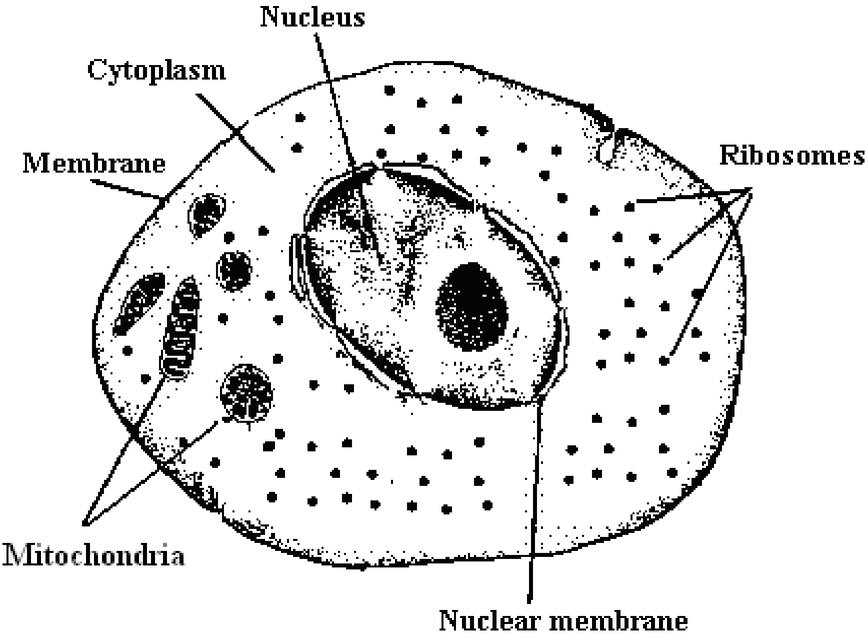


Fig. 1.2 Eukaryotic cell

thus comprise the metabolic machinery of the cell. For example, adenosine triphosphate (ATP), the main source of the chemical energy of the cell, is produced either within *mitochondria* in the process of cellular "respiration" or within *chloroplasts* during photosynthesis. The two organelles are called, therefore, the "power plants" of the cell.

1.2 Molecular Basis of Heredity - DNA and Complementary Nucleotides

1.2.1 The Double Helix

Nucleic acids, DNA and RNA, are organic compounds, which play the central role in the cell since they contain all the information required to duplicate and maintain the organism. The universal basis of heredity consists in synthesizing exact copies of the parent DNA (or RNA), which are transmitted from one generation to the next. The universal mechanisms directing the processes of protein[3] and RNA synthesis and thus regulating the cell's activities, also

[3] **Proteins** are large organic molecules composed of amino acids polymerized in a linear chain, which, in turn, forms more compact higher-order spatial structures. Proteins act as enzymes, structural units, and regulators of metabolic processes in a cell.

include information transmission from DNA, serving as a template for RNA synthesis. Thus, in order to understand the basics of the information flow in a cell, we should get acquainted with two main types of nucleic acids.

DNA (deoxyribonucleic acid) and *RNA* (ribonucleic acid) represent long chains of repeating monomers[4], called *nucleotides*. Nucleotides are constructed of sugar molecules (ribose or deoxyribose in the case of RNA or DNA, respectively) and phosphate groups, which make up the chain backbone, and of *bases*, which are attached to sugars. Three bases are pyrimidine derivatives: cytosine (C), thymine (T) (found in DNA), and uracil (U) (found in RNA); their single-ring structures are shown in Fig. 1.3A. Two bases, adenine (A) and guanine (G), are purine derivatives, which have a two-ring structure (Fig. 1.3B). Although the “building blocks” of DNA and RNA are almost the same, the two molecules differ in their structure and function

In eukaryotes, bacteria, and in most viruses, DNA that stores genetic information has the structure of a *double helix*; its canonical form is called B-DNA. In 1953, Watson and Crick published their famous article [281], where they suggested the structure of B-DNA as a right-handed helix formed by sugar-phosphate backbones of two individual DNA strands, which are wound around each other. Because of the chemical structure of the deoxyribose residue, the two ends of the DNA strand are not equivalent – at one end, the $3'$-end, there is an exposed OH- group on the deoxyribose, while at the other end, the $5'$-end, there is an exposed phosphate group. In the double helix, the two strands are *antiparallel*, one strand having the $5' - 3'$ direction and the complementary strand having the $3' - 5'$ direction. We will see further that the notion of the DNA (RNA) strand <u>direction</u> is essential in the processes of DNA or RNA synthesis in a cell.

An idealized model of the DNA double helix is shown in Fig. 1.4. It can be seen that the bases which belong to different strands and are situated opposite to each other form *base pairs*, stabilized by hydrogen bonds. The vertical distance between two nearest stacked base pairs is about 3.4 angstroms and one helical turn contains approximately 10 base pairs. The width of the helix is 20 angstroms.

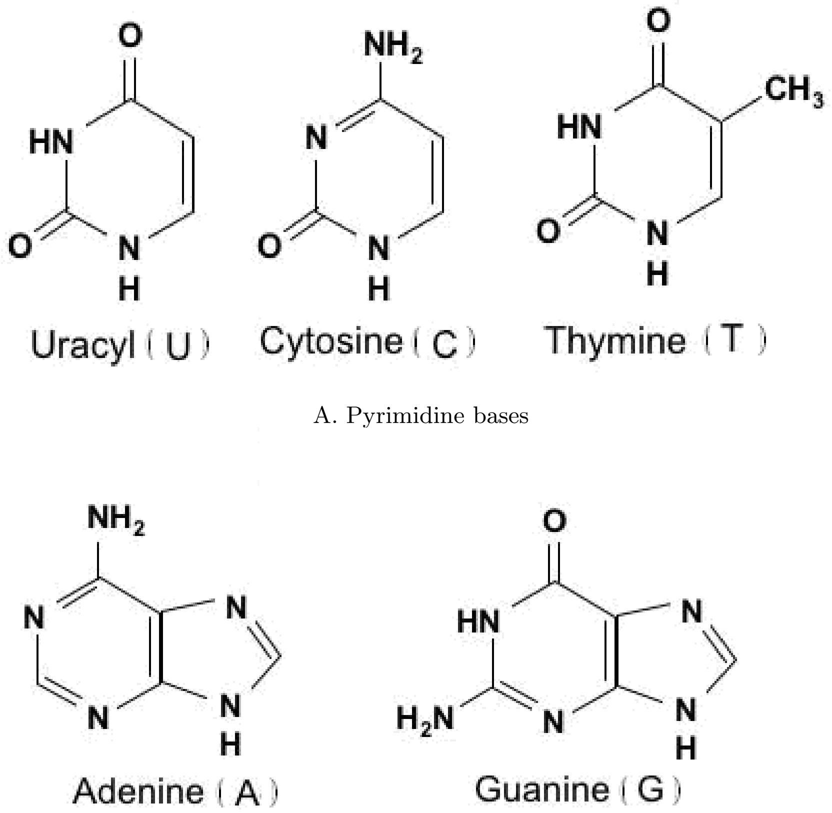
The base pairs within the DNA double helix are formed in a very specific way. Since the shapes and electrical charges of purines and pyrimidines are *complementary*[5], A binds only with T (U), while C binds only with G (see Fig. 1.5).

1.2.1.1 Chargaff’s Rules

The Chargaff’s Parity Rule 1 states that, in a double-stranded DNA molecule from any cell of any organism, the content of guanine equals that of cytosine,

[4] A **monomer** (from Greek *mono* “one” and *meros* “part”) is a small molecule that may become chemically bonded to other monomers to form a **polymer**.

[5] **Complementary** means “matching opposite”.



A. Pyrimidine bases

B.

Purine bases

Fig. 1.3 Chemical structure of nucleic acid bases

while the content of adenine equals that of thymine [42]. Obviously, this rule is a direct consequence of the complementary base-pairing in DNA described above.

The Chargaff's Parity Rule 2 states that complementary nucleotides are met with almost equal frequencies in each of the two DNA strands [43]. In other words, sufficiently long (> 100 Kb[6]) natural DNA sequence contains approximately equal amounts of guanine and cytosine and approximately equal amounts of adenine and thymine.

[6] The length of a single-stranded nucleic acid is measured in the number of nucleotides. The length of a double-stranded sequence is measured in base pairs (**bp**), thousands of base pairs (**kilobases**, **Kb**), millions of base pairs (**megabases**, **Mb**), or billions of base pairs (**gigabases**, **Gb**).

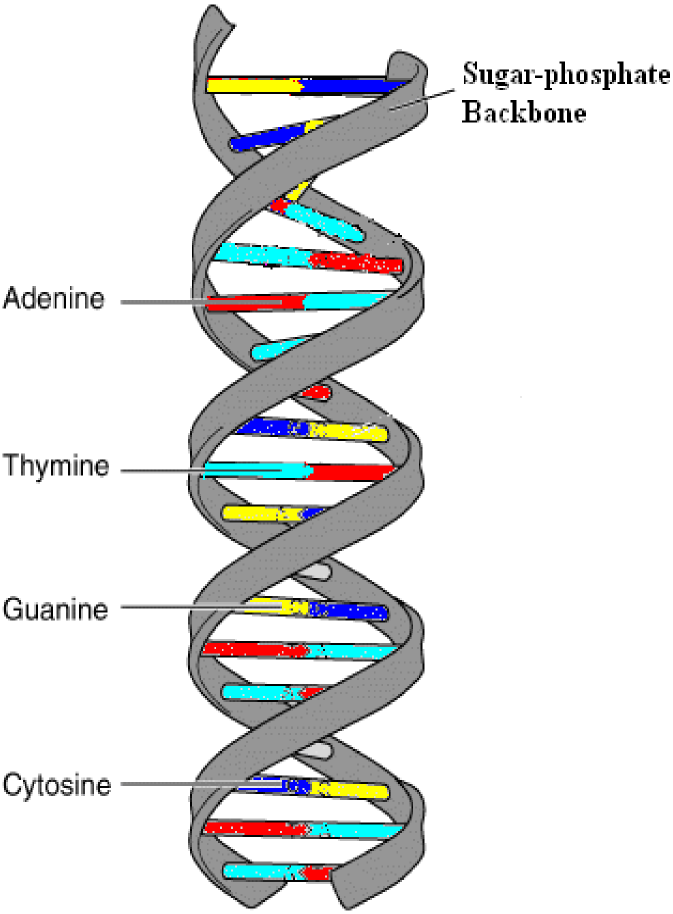


Fig. 1.4 DNA double helix

It is known that relative amounts of A, G, T, and C bases in DNA molecules from different species may be quite different. According to the Chargaff's Rule 2, it is sufficient to know the content of one nucleotide in a long genomic sequence in order to calculate the content of any of the other three nucleotides.

This rule was shown to hold for DNA fragments from eukaryotic, eubacteria, archaebacteria genomes [291], [189], [192], and also for oligonucleotides[7] - it was found that if a sufficiently long ($>$ 100 Kb) strand of genomic DNA contains N copies of an oligonucleotide, it also contains N copies of the reverse-complementary oligonucleotide (Albrecht-Buehler, 2006).

[7] **Oligonucleotides** are short sequences of nucleotides.

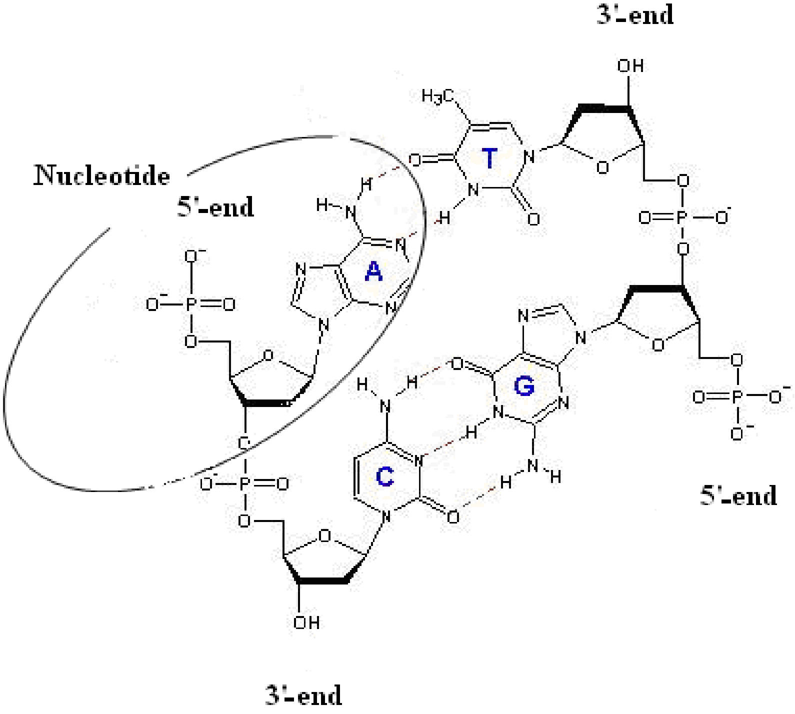


Fig. 1.5 Complementary base pairs in DNA

1.3 Functional Structure of DNA within the Cell

1.3.1 Genes

A *gene* is a segment of DNA (or, in some viruses, RNA) that has specific linear sequence of nucleotide bases and a certain definite location on one strand of DNA (RNA). When the gene is in an active state (is being expressed), a complementary RNA segment is synthesized on one strand of the DNA helix; this process is called *transcription* (see Sections 1.3.2 and 1.4.2). The resulting single-stranded RNA molecules may further participate in protein synthesis, carrying information about the amino-acid sequences (see footnote 3 above). Alternatively, RNA molecules may be used as regulatory or structural elements of the cell. For example, there exist several types of ribosomal RNA molecules (rRNA), which, being incorporated in ribosomes, direct protein synthesis, or about 20 types of transport RNA (tRNA) molecules, which supply specific amino acids to the ribosome (see Section 1.4.2.3 and Fig. 1.10). Accordingly, genes are classified as *protein-coding* and *RNA genes*. Thus a *gene* may be defined as a segment of DNA sequence corresponding to one or more proteins or to a single catalytic or structural RNA molecule.

Below, we describe the structures of prokaryotic and eukaryotic genes in "functional" terms, that is, with respect to the process of transcription.

1.3.2 Structure of Genes

1.3.2.1 Prokaryotic Gene

There are two essential facts that one should know about the process of nucleic acid synthesis in a cell. First, for chemical reasons, the synthesis always occurs in the $5'-3'$ direction. Second, a complementary RNA molecule is synthesized only on one strand of the double helix, which is called the *template* strand.

Schematic structure of the prokaryotic gene is shown in Fig. 1.6. The process of transcription is initiated in the region called *promoter*. The first region of the promoter that is recognized by the enzyme *DNA-dependent RNA polymerase* is centered at the distance of about 35 bp before the start of transcription. Since the $5'-3'$ direction is referred to as *upstream* (the $3'-5'$ direction being called *downstream*), we can say that it is located at -35 bp upstream from the start. In those cases when the nucleotide sequence of a DNA region varies, it is often possible to determine the so-called *consensus sequence* with the highest frequencies of nucleotide occurrences in each position. The consensus sequence of the -35 bp region is TTGACAT. When the RNA polymerase binds to both strands of DNA at this site, the transcription is initiated. Next, the enzyme, moving along the DNA, finds the so-called *Pribnow box* region (with the TATAAT consensus sequence), centered at about -10 bp upstream from the start. Here the two DNA strands are separated and RNA synthesis starts on the template strand at the position of about 7 bp downstream from the Pribnow box. The process of templated polymerization guided by the RNA polymerase consists in the synthesis of the RNA chain where each subsequent nucleotide base is complementary to the corresponding base on the template DNA strand. As a result, in the transcribed fragment, the sequence of the four DNA bases (A, C, G, and T) is preserved in the sequence of the four RNA bases (U, G, C, and A), respectively. The polymerization proceeds until the RNA polymerase meets the stop (*terminator*) signal. In prokaryotes, two types of transcription terminators have been discovered, one of them being a (G,C)-rich sequence with the secondary structure of a *hairpin loop* (see Fig. 1.6), which acts as a hindrance for the RNA polymerase further movement. The part of the DNA sequence between the promoter and the terminator is, actually, the gene. The length of prokaryotic genes ranges from about 250 bp to as much as 100,000 bp.

1.3.2.2 Eukaryotic Gene

Schematic representation of a eukaryotic protein-coding gene is shown in Fig. 1.7. It can be immediately seen that the gene's structure is much more

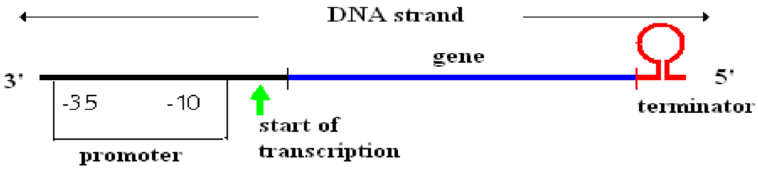


Fig. 1.6 Schematic representation of the prokaryotic gene structure

complicated than that of a prokaryotic gene. First of all, eukaryotic genes are not directly accessible to the RNA polymerase due to the compact structure of the DNA-histone complex in chromatin (see Section 1.5). Several proteins, the so-called *transcription factors,* which are attached to different promoter regions, help the enzyme find its binding site on the promoter. Since, besides the RNA polymerase, quite a few proteins are assembled on the promoter, the latter has a complex structure, which includes the *core (basal) promoter* (the TATA box, located about -30 bp upstream from the start of transcription) and several upstream regions (the CAAT and GC boxes, which may be located as far as -200 bp upstream). After the *preinitiation complex* of the transcription factors with the RNA polymerase is assembled on the promoter and the DNA strands are separated, the transcription is initiated. However, still other proteins such as *enhancers*, which may be located far away along the DNA strand, are required to control the efficiency and the rate of the process. According to the universal mechanism of information flow from DNA to RNA, the RNA polymerase moves along the coding DNA strand, using it as a template for unidirectional ($5' - 3'$ or downstream) polymerization of nucleotides, guided by their structural and chemical complementarity. However, while in prokaryotes the coding part of the gene is a continuous sequence, the transcribed part of a eukaryotic gene consists of *exons*, separated by non-coding DNA stretches, called *introns.* The $3'$-end of the first exon serves as the *transcription initiation site,* while at the $5'$-end of the gene the transcription *termination signals* are located. The RNA *primary transcript* (*pre-mRNA*) undergoes a few steps of processing, in particular, *splicing*, which results in the removal of introns. The mature mRNA contains the exons, and the total of those exons or their parts that are translated into a protein sequence are referred to as the *coding part* of the gene.

The length range of eukaryotic genes is much larger than that of prokaryotes. For example, the length of histone genes is about 1,000 bp, while that of the dystrophine gene is 2,500,000 bp.

1.3.3 Non-coding DNA

Natural DNA molecules usually contain a great number of genes aligned on the DNA strand. However, the genes do not have the end-to-end alignment,

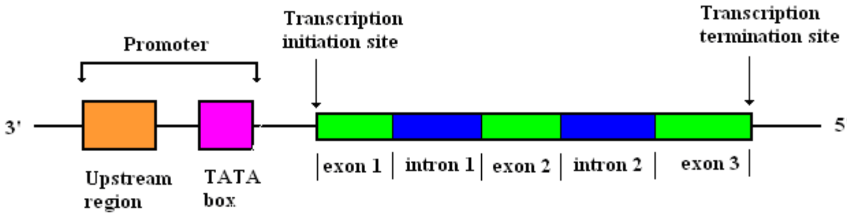


Fig. 1.7 Schematic representation of the structure of a eukaryotic protein-coding gene

being separated by intergenic sequences of various lengths. The DNA in these sequences is referred to as *non-coding* DNA. In eukaryotic organisms, this term relates both to the intergenic regions and to introns located within genes. The fraction of non-coding DNA is different for various species, e.g., for most bacteria, it is about 10-20% of the total DNA length in the cell, while in humans it amounts to as much as about 95%. It has been established by now that non-coding DNA comprises:

different types of *regulatory regions* such as the above-mentioned enhancers;

transposons - DNA sequences that can move to different positions within a single cell, in particular, causing mutations. In eukaryotes, transposons amount to a large fraction of the total DNA;

pseudogenes - sequences remaining from ancient genes which have lost their protein-coding function;

introns - although these sequences are not translated, they were shown to have an effect on the produced proteins;

virus relics that once joined the cell DNA and have been conserved ever since.

Although the non-coding DNA is being extensively studied, the function of most of its part is not known yet.

1.4 Protein Synthesis in a Living Cell

1.4.1 Genetic Code

Now let us return to the conception of the DNA as a sequence of nucleotides, which may contain one of the four bases - A, C, T, or G. Each nucleotide can be regarded as a letter in a four-letter alphabet so that the DNA strand may be viewed as a sequence of letters, i.e., as a text. As we have already seen above, the information required for the synthesis of protein molecules is written in those parts of the text which correspond to genes. The building blocks of a protein chain are amino-acid residues. The establishment of the *genetic code*, i.e., the way in which the sequence of amino acids constituting a protein is encoded in the sequence of DNA (and, thus mRNA) nucleotides was a major breakthrough of the 20^{th}-century molecular biology. It turned out that the elementary unit that encodes one amino acid - *codon* - is a sequence of three nucleotides. Obviously, if a nucleotide sequence is viewed as a text, then a codon, being the elementary chunk of information, should be considered as a word. The maximum possible number of different words in the DNA "language" is $4^4 = 64$ (it was shown that all of them are code words). Taking into account the existence of twenty different amino acids, it can be concluded that one amino acid may be encoded by more than one codon, which is, actually, the case. There also exist start and stop codons, which designate the beginning and the end of the amino-acid chain synthesis. The genetic code is the same in most organisms; yet it is not universal since in some cases (e.g., for the mitochondrial DNA) the code differs from the *canonical* (*standard* one (see Table 8.1).

Table 1.1 The Genetic Code

Amino Acid	**mRNA codons**
Alanine (Ala)	GCA, GCC, GCG, GCU
Arginine (Arg)	AGA, AGG, CGA, CGC, CGG, CGU
Asparagine (Asn)	AAC, AAU
Aspartic acid (Asp)	GAC, GAU
Cysteine (Cys)	UGC, UGU
Glutamic acid (Glu)	GAA,GAG
Glutamine (Gln)	CAA,CAG
Glycine (Gly)	GGA, GGC, GGG, GGU
Histidine (His)	CAC, CAU
Isoleucine (Ile)	AUA, AUC, AUU
Leucine (Leu)	CUA, CUC, CUG, CUU, UUA, UUG
Lysine (Lis)	AAA, AAG
Methionine (Met)	AUG*
Phenylalanine (Phe)	UUC, UUU
Proline (Pro)	CCA,CCC,CCG, CCU
Serine (Ser)	AGC, AGU, UCA, UCC, UCG, UCU
Threonine (Thr)	ACA, ACC, ACG, ACU
Tryptophan (Trp)	UGG
Tyrosine (Tyr)	UAC, UAU
Valine (Val)	GUA, GUC, GUG, GUU
Stop codons	UAA, UAG, UGA
*) At the beginning of a gene, AUG has the function of the start codon.	

1.4.2 Flow of Genetic Information in the Cell

1.4.2.1 The Central Dogma of Molecular Biology

If both the DNA and the protein sequences are viewed as texts or, in other words, as messages containing certain information, the question arises as to the directions in which this information may be transferred within the cell. The answer to this question was formulated by Francis Crick as a fundamental principle (the so-called *central dogma*) of molecular biology. According to this principle, the possible directions of information transfer are from nucleic acid to nucleic acid or from nucleic acid to protein, but not from protein to protein or from protein to nucleic acid (see the scheme in Fig. 1.8).

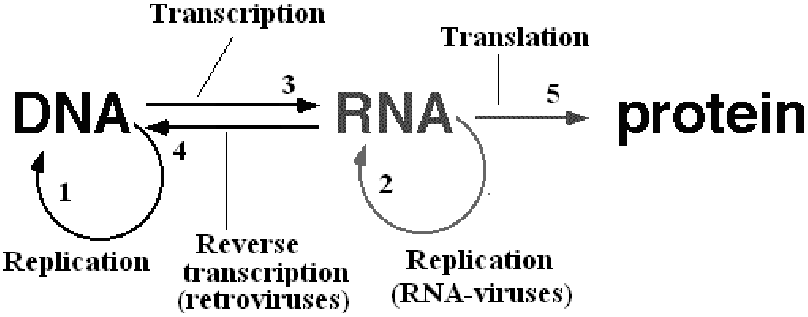

Transcription
Translation
DNA
3
RNA
5
protein
4
1
2
Replication
Reverse transcription (retroviruses)
Replication (RNA-viruses)

Fig. 1.8 Scheme illustrating the central dogma of molecular biology

In other words, this means that a nucleotide sequence can determine the sequence of another nucleic acid or of a protein, but the sequence of amino acids cannot determine the sequence of another protein or of a nucleic acid.

1.4.2.2 Information Transfer from Nucleic Acid to Nucleic Acid

The process in which DNA serves as a template to reproduce itself is called *replication* (see Fig. 1.8, arrow 1). The two DNA strands are unwound and, on each of them, a new strand is synthesized by the DNA polymerase enzyme, which adds subsequent nucleotides complementary to those on the template strand. In this way, two exact copies of the original double-stranded DNA molecule are produced, which is the molecular basis of the amazingly conservative phenomenon of *heredity* - transmission of genetic information from ancestors to descendants.

It has been mentioned above that some viruses (e.g., those causing hepatitis C and influenza) contain RNA as the store of genetic information. When such virus infects a cell, it reproduces itself (replicates) through RNA synthesis on the RNA template (Fig. 1.8, arrow 2). This process is catalyzed by the *RNA-dependent RNA polymerase* (called also *RNA replicase)* enzyme.

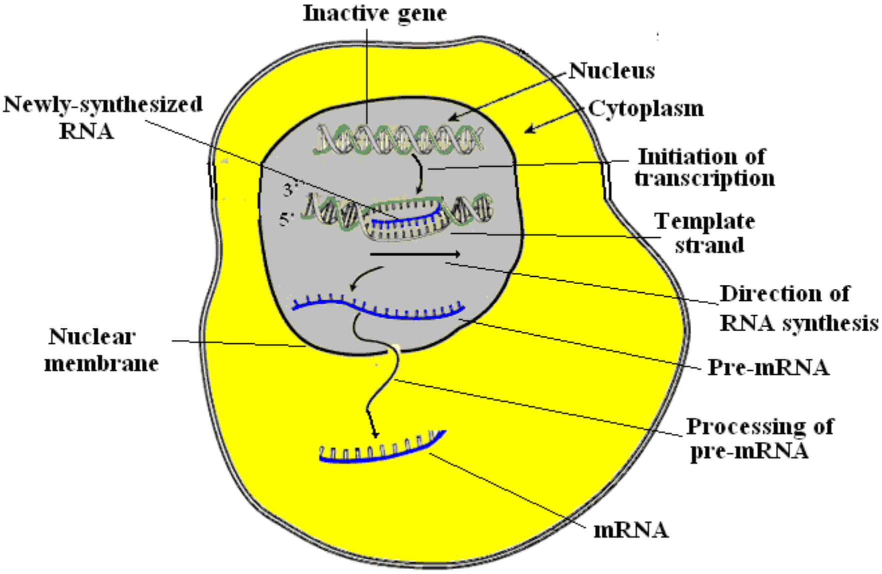


Fig. 1.9 Transcription

The process of *transcription* (Fig. 1.8, arrow 3), in which an RNA molecule is synthesized on the DNA template, has already been described above (see Sections 1.3.2 and 1.4.2 and Fig. 1.9. The subsequent processing of the produced RNA molecules may result in the formation of a messenger RNA (mRNA), which further serves as a template for either the target-protein synthesis, or structural and regulatory RNA such as ribosomal RNA (rRNA) or transport RNA (tRNA). There also exists a group of RNA viruses which are not capable of RNA replication. When the virus infects the host cell, it uses the enzyme called *reverse transcriptase* to synthesize DNA using its RNA genome as a template. The DNA which is produced via such *reverse transcription* is integrated into the host's genome, after which the virus can replicate as part of the host cell's DNA.

1.4.2.3 Information Transfer from Nucleic Acid to Protein

The process in which proteins are built according to the information encoded in the DNA (RNA) sequence can be described, in linguistic terms, as translation of the nucleic acid text written by means of the nucleotide four-letter alphabet into the protein text written by means of the amino-acid twenty-letter alphabet. For this reason, the process of protein synthesis on the mRNA template is called *translation*. The general features of this process are the same for all species on Earth.

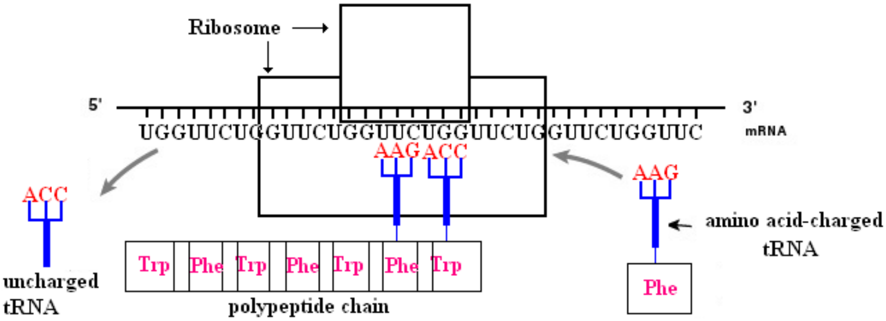


Fig. 1.10 Translation

The DNA coding regions are not directly used as templates for protein synthesis – instead, an intermediate molecule, which has the same sequence of codons, is transcribed from DNA. The mature form of the transcript, messenger RNA (mRNA), comes out of the nucleus into the cytoplasm and gets attached to special organelles, *ribosomes*. Ribosomes may be called “molecular machines”, which take, as an input, an mRNA molecule and produce, as an output, a sequence of amino acids (a protein chain). The number of ribosomes in a cell ranges from thousands in bacteria to hundreds of thousands in human cells. mRNA molecules bind between the two ribosome subunits, which are composed of proteins and several types of rRNA molecules, the former having mainly structural functions and the latter catalyzing the process of translation. The codons on the mRNA do not overlap and they are sequentially read from the start codon to the stop codon in the $5' - 3'$ direction. Each consequent amino acid is put in its right position with the aid of an adaptor molecule, *transfer RNA* (*tRNA*). Since there exist 20 amino acids and still more codons (see above), there also exists a whole family of tRNAs, each kind of tRNA having affinity with and thus binding to its specific codon on mRNA. On the other hand, each type of tRNA has the corresponding amino acid attached to the opposite part of the molecule. As the ribosome moves along mRNA, specific tRNAs bind to the consecutive codons and their attached amino acids are polymerized to form the encoded amino-acid chain.

1.5 Chromosome

In a cell, DNA molecules exist in the form of *chromosomes*, which are mainly localized in the cell’s nucleus. A eukaryotic chromosome is a complex of one continuous DNA molecule with numerous copies of several types of *histone proteins* (or *histones*) and with many different non-histone proteins, most of which are transcription factors.

Table 1.2 Diploid number of chromosomes in some species

Species	**Diploid number of chromosomes**
Rye	14
Human	46
Goldfish	104

In most prokaryotes, a single circular chromosome is organized by proteins in a distinct structure within a restricted region, called *nucleoid.* Some prokaryotes and a few eukaryotes have DNA molecules outside the chromosomes in the form of *plasmids* - circular, double-stranded DNA fragments, which replicate within a cell independently of the chromosomal DNA. Mitochondria in eukaryotic cells also have their own DNA, packed in a chromosome.

DNA molecules in chromosomes are very long. For example, in prokaryotes, their length may range between a hundred to ten thousand kilobases, which would correspond to the length of a stretched DNA of about 10^{-3}–10^{-1}cm. DNA in eukaryotic chromosomes is still longer - for example, the length of DNA is about 5×10^7 bp in the smallest human chromosome and 2.5×10^8 bp in the largest, which corresponds to the length of an extended DNA molecule 1.7 and 8.5 cm, respectively. It would be most impressive to imagine that if all the DNA in a single human cell were stretched end-to-end, it would have the length of 2 m! Taking into account the average prokaryotic and eukaryotic cell dimensions (see Section 1.1), one can understand why chromosomes have a perfectly compact ordered (supercoiled) structure, maintained by histones (or, in prokaryotes, by histone-like proteins). The degree of chromosome condensation varies with the cell's cycle, reaching its maximum in the phase of cell division, when individual chromosomes are visible in an optical microscope. Historically, these had been discovered long before the DNA era and, for the sake of convenience, designated by numbers or letters. Later, these chromosome identifiers were also applied to the corresponding DNA molecules. Thus one should keep in mind that the phrase "a gene in chromosome 3" refers to the gene in the DNA molecule which, together with proteins, forms chromosome 3.

The number of different chromosomes in the nucleus varies from one to more than a hundred for different species (see Table 1.2). Most bacteria have one main chromosome and a variable number of plasmids and their copies (in some cases, the single main chromosome may also exist in multiple copies). Those species which reproduce asexually have the same, specific for each species, set of chromosomes in each cell of the organism. The same is true for reproductive cells of sexually reproducing species (including humans), while in all the other (somatic) cells each chromosome has its pair, so that the set is doubled (a diploid set).

1.6 Genome, Proteome, and Phenome

If we consider DNA molecules as nucleotide sequences, it becomes clear that the same chromosomes in two different organisms of the same species do not contain identical DNA sequences. Indeed, in a particular organism, each gene can be represented by one of its possible variations (*alleles*), having varied nucleotide sequences. Nevertheless, the total set of genes, which includes all types of alleles, can be considered to be constant for the same species and it is this set that is usually referred to as the species' *genome* (e.g., human genome). Sometimes, this term is used in the (narrower) sense of the set of all genes of a particular organism. In this book, we employ the first of the above genome definitions.

In different species, genome sizes can differ considerably. For example, the size of the *H. influenzae* bacterium genome is about 1.8x10^6 bp (for different bacteria, this value lies within the approximate limits of 0.5-5x10^6 bp), while the *Homo sapiens* (human) genome size is as large as 3.2x10^9 bp. The number of genes in the genome also varies, though not so dramatically - in our example, the number of genes is about 1,700 in the *H. influenzae* genome and 25,000 in the *Homo sapiens* genome (in bacteria, the total number of genes may range from 0.5x10^3) to 4.3x10^3).

According to Norman G. Anderson [12], genes give the general plan of the cell's structure and function, while proteins determine the actual active life of the cell. In other words, life is realized at the level of proteins. The estimated number of different proteins in a human body is about 300,000 or more, which is 10 times larger than the total number of genes. This is due to variations in the mRNA sequence which are introduced during the pre-mRNA processing (as a result, a single gene may, actually, code for a variety of proteins) and to post-translational modifications of proteins. Consider also the innumerable interactions between the proteins and one can imagine the measure of complexity of the cell's (organism's) protein system. To deal with such systems, the notion of *proteome* is introduced by analogy with that of genome. There exist a few proteome types, in particular, a *cellular proteome* (the set of proteins in a particular cell type under particular conditions) or a *complete proteome* (the complete set of proteins composed of various cellular proteomes).

As we have seen above, each particular organism of the same species has a different genome and thus a different complete proteome, which results in a variety of traits that are characteristic of different individuals. A limited set of observable traits (such as height, color of eyes, or blood group) is usually called a *phenotype.* The set of all organism's traits constitutes another level of an organism's description, which is referred to as *phenome.*

In conclusion, just as the genome and proteome relate to the sets of an organism's genes and proteins, respectively, the phenome relates to the complete set of an organism's phenotypic traits.

Chapter 2
Biological Classification

It is no secret that people are fond of classifying everything. Biology is no exception. It is a lot easier to study living beings if we have a system that separates them into specific groups. There exist a lot of characteristics and methods which can be used for the purpose of classification. In this book, we will discuss a number of classification methods based on genome sequences. However, before we start describing these methods, it may be useful to illustrate a few classic approaches to classification by the example of microorganisms.

Example 1. All prokaryotes may be grouped on the basis of their optimal growth temperature (see Fig. 2.1). Four temperature groups are defined in the literature:

- *Psychrophiles* - organisms that are able to grow at temperatures below 0°C;
- *Mesophiles* - organisms that grow best between 10 and 30°C;
- *Thermophiles* - organisms that grow best in hot conditions, between 30 and 50°C;
- *Hyperthermophiles* - organisms that thrive in environments hotter than 60°C, the optimal growth temperatures ranging from 80 to 110°C.

Example 2. Another criterion for classifying bacteria may be their belonging to the aerobic or anaerobic type. Those bacteria that require free oxygen to support life are refered to as *aerobic.* Those bacteria that are able to live in the absence of oxygen are called *anaerobic.* The latter are, in turn, divided into the following three groups:

- *Obligate anaerobes* will die when exposed to atmospheric levels of oxygen;
- *Facultative anaerobes* can use oxygen when it is present;
- *Aerotolerant* organisms survive in the presence of oxygen, but they do not use it as a terminal electron acceptor.

Example 3. Still other example is the empirical Gram classification system, which is largely based on the differences in the structure of cell walls. These

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 17–22, 2010.
springerlink.com

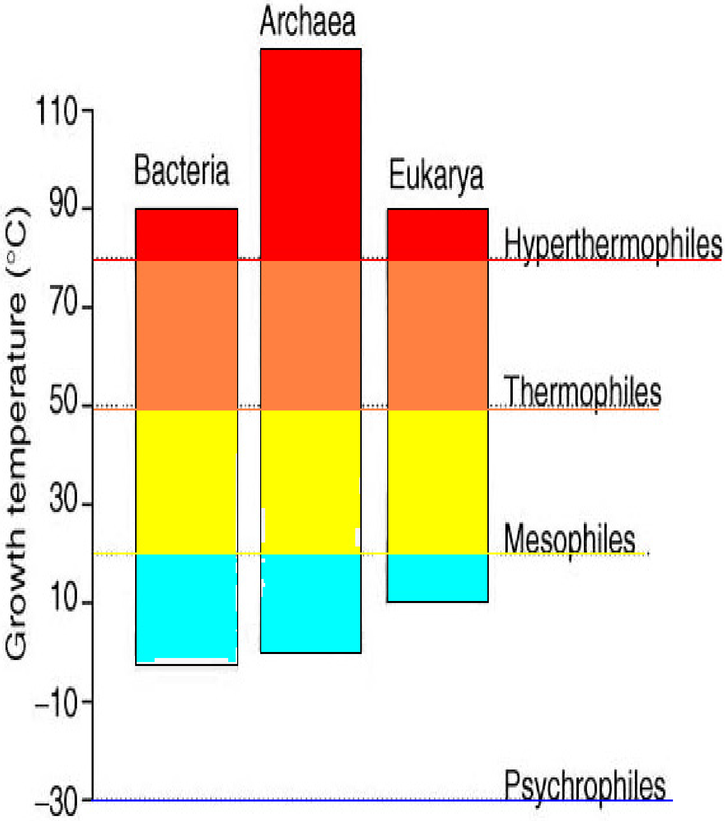


Fig. 2.1 Classification of species according to their growth temperature

differences can be visualized using the technique of Gram staining. Gram-positive bacteria appear blue or violet under a microscope, while Gram-negative bacteria appear red or pink.

2.1 Biological Systematics

Biological systematics studies the relationships of species starting from the origin of life on Earth up to the present days. Obviously, prior to studying such relationships, it is crucial to be able to properly describe these organisms themselves. Just this is the objective of *taxonomy*, the science which describes, identifies, classifies living beings and gives them appropriate names.

The Swedish biologist Carl Linnaeus, who lived in the 18th century, is considered to be the founder of modern taxonomy. The Linnaean classification system is organized according to a hierarchical principle and comprises the following main *taxonomic ranks* or *taxa* (singular *taxon*):

1) Kingdom; 2) Phylum (division); 3) Class; 4) Order; 5) Family; 6) Genus; 7) Species.

To identify each species, Linnaeus introduced binomial nomenclature, which uses the combination of the genus and the species name. According to the current rules,

- Names must be in Latin and printed in italics.
- The genus name (*Homo* in *Homo sapiens*) is capitalized and must be a single word.
- The species name (*sapiens* in *Homo sapiens*) can be either a single word or a compound word.

For example, *Escherichia coli* is one of several types of bacteria that normally inhabit the intestine of humans; *Helicobacter pylori* is a human gastric pathogen, causing peptic ulcers; *Mus musculus* is the common house mouse.

Originally, two *Kingdoms* of living beings were distinguished by Linnaeus: *animals* (*Animalia*) and *plants* (*Vegetabilia*). In the next classification system, the highest rank of *Domain* was introduced, while Kingdom became a subdivision of Domain. The first two-domain system classified life into *Prokaryota* and *Eukaryota*. In 1990, Carl Woese suggested a three-domain system, dividing all cellular forms of life into *Archaea*, *Bacteria*, and *Eukarya* domains. Currently, *Archaea* and *Bacteria* are viewed as single-kingdom domains, while *Eukarya* is subdivided into four kingdoms. The resulting six-kingdom system (*Animalia*, *Plantae*, *Fungi*, *Protista*, *Archaebacteria*, and *Eubacteria*) is widely used in the United States (Fig. 2.2). However, many biologists still employ the five-kingdom system (proposed by R. Whittaker in 1969), where *Archaebacteria* and *Eubacteria* constitute one *Prokaryota* kingdom. Obviously, this system is based on just two domains - *Eukarya* and *Prokaryota*.

2.2 Phylogenetics

Phylogenetics[1] is the study of evolutionary relationships among various groups of organisms. In other words, this branch of biology deals with the evolutionary history of living organisms and its results are usually presented in the form of the so-called *evolutionary* or *phylogenetic trees*. Before the Genomic Era, such trees were constructed by studying, analyzing, and comparing all kinds of organisms' traits. In our days, phylogenetic trees are built on the basis of the data on genomic sequences. Various types of molecular biological sequence have been used to reveal the phylogenetic relationships "across life".

The first phylogenetic trees, which were built on the basis of DNA, RNA, and protein relatively small conserved sequences [284], [285], enabled the

[1] The word is of Greek origine: *phyle* - tribe, race + *genesis* - relative to birth.

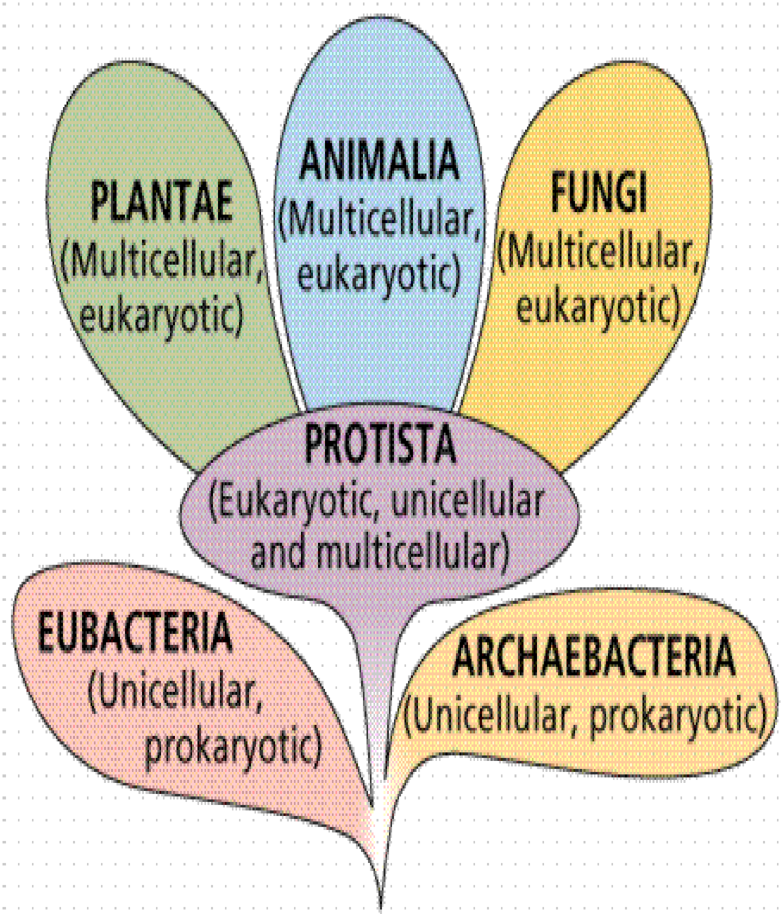

PLANTAE
(Multicellular, eukaryotic)
ANIMALIA
(Multicellular, eukaryotic)
FUNGI
(Multicellular, eukaryotic)
PROTISTA
(Eukaryotic, unicellular and multicellular)
EUBACTERIA
(Unicellular, prokaryotic)
ARCHAEBACTERIA
(Unicellular, prokaryotic)

Fig. 2.2 A schematic representation of the three-domain (six-kingdom) phylogenetic classification system. (Image from: Purves *et al.* Life: The Science of Biology, fourth edition. Sinauer Associates and WH Freeman)

researchers to establish the *Universal-Tree-of-Life* concept (Fig.2.3). Such trees demonstrate the evolutionary process in the direction of the current diversity of life. Each node in the tree corresponds, actually, to a taxonomic unit and represents the most recent common ancestor for the species which diverge from the node.

The overall structure of the Universal Tree of Life corresponds to the three-domain classification system described above (see also Fig. 2.2). However, it was found out later that, if various genome characteristics, such as specific genes, rRNA-coding sequences, intergenic spacers, and even the number of genes in the genome, were taken as a reference for taxonomic reconstruction, quite different taxonomic relationships could be obtained [32], [33], [286]. For

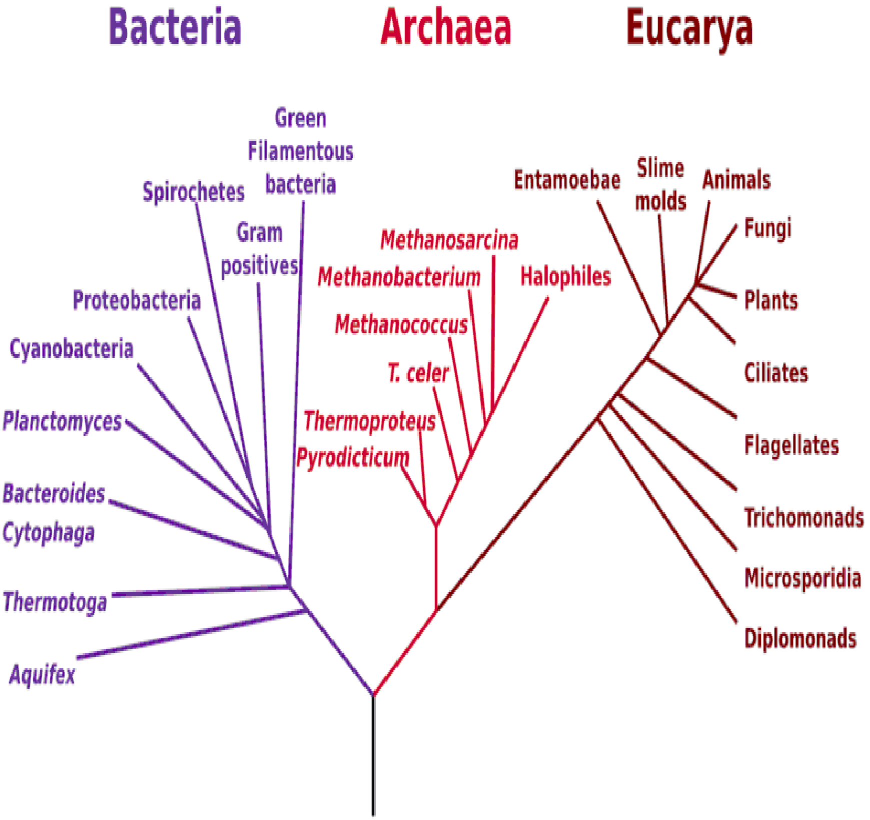


Fig. 2.3 Phylogenetic tree of life

example, Tekaia [246] assessed common ancestry by comparing the contents of whole genomes, their overall gene similarities, and the loss or acquisition of genes. The phenograms obtained in this work showed remarkable correspondence with the standard ribosomal phylogeny. By contrast, Golding and Gupta [91], on the basis of seven protein-coding genes, obtained a phylogenetic tree which could not be explained by the traditional three-domain scheme. Lin and Gerstein [172] built whole-genome trees based on the presence or absence of particular molecular features in the genomes of a number of microorganisms. The authors found that their phenograms agreed fairly well with the traditional ribosomal phylogeny. However, phylogenetic distribution for certain classes of protein folds, e.g., all-beta ones, was quite different (see Chapter 7 for further discussion).

It should be noted that the efforts to classify organisms on the basis of phenotypical traits do not always result in adequate phylogeny, either. The closeness of phenotypical traits of some species appears to be connected not so much with their evolutionary closeness, but rather with the convergence on the basis of a similar function or, speaking more generally, of a similar ecological pressure. There exist numerous examples of such common-function-based similarity, one example being a far-gone convergence between such taxonomically distant species as *Cephalopoda* and *Fish*. The convergence is manifested not only through structure and physiology, but also through the role of these species in the ecology and general biology of the ocean. In other words, as E. Packard remarked, *"Functionally, cephalopoda are fish" [194].* The observed dissimilarities in classifications based on the comparison of various genome parameters may be interpreted in the same way. Namely, if the resulting classification can be assessed as a "non-phylogenic", it can be suggested that its fundamental parameters are prone not to the evolutionary, but to other types of changes. In the case of genomes, apart from the "ecological factor", the direct exchange of genetic material (parallel transfer) should also be taken into account.

Chapter 3
Mathematical Models for the Analysis of Natural-Language Documents

In this chapter, we describe some models for studying natural-language documents which are useful for the analysis of DNA texts. Although most models for analyzing natural languages had appeared before molecular texts could be decoded, these models were not always employed directly, but rather discovered afresh. The relevance of each model for language and molecular text analysis may be different. Nevertheless, in order to properly understand the ideas underlying the models, it is helpful to introduce, first, the models for the analysis of natural-language documents.

3.1 Direct Comparison of Texts

3.1.1 Distance between Two Strings

The concept of the *distance between two strings* can be applied to a) the search and processing of records in search engines based on the record names; b) the search in database with an incomplete or ambiguous preset name; c) the correction of input errors or those which appear in the course of automatic recognition of scanned documents or recorded speech, and d) other tasks connected with automatic text processing.

The general basic idea is defining the distance between two strings as the number (perhaps, the weighted number) of specific operations which transform one string into the other. The particular set of these operations depends on the type of the process which generates "close" strings. For example, if the text is being typed, the dominant mistakes will be deletions or insertions of letters, substitution of one character for another, and transpositions of adjacent characters. In this case, the above-mentioned "specific operations" involved in constructing the distance between the correct and the typed texts would be all the sources of mistakes. The distance defined in such a way is called the *Damerau-Levenshtein distance* [55].

Consider the following example. The words *distance* and *distanse* differ only in one substitution (s is substituted for c), thus the distance between

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 23–42, 2010.
springerlink.com

the words equals 1. The words *distance* and *ditsanse* differ in one substitution and one transposition; as a result, the distance between the words is 2 if the mistake weights are 1.

If transpositions are not included in the set of specific operations, the resulting distance between two strings is referred to as *Levenshtein distance* [167], which was introduced for binary codes. In contrast to this, the *Jaro-Winkler distance* [283] is a measure of dissimilarity between two strings which predominantly considers the number of transpositions in a string.

3.1.2 Text Identification as an Authorship Attribution Problem

Authorship attribution is the task of identifying the author of a given text. It can be considered as a typical classification problem, in which a set of documents with known authorship is used for training, the final goal being automatic determination of the author of an anonymous text. In contrast to other classification tasks, it is not clear which features of a text should be used to identify the author. Consequently, the main concern of the computer-assisted authorship attribution problem is defining such evaluation of documents which would unambiguously characterize the authors' writing styles. Note that it is the style peculiarities and not the specific topic of the document which is essential for such type of classification. Therefore, different formal parameters have to be suggested for the authorship characterization. For example, the *Markov chains* (originally referred to as "trials linked in a chain") were first used [182] to determine the peculiarities of vowel and consonant distributions in the poem "Evgeny Onegin" by A.S. Pushkin. Currently, the methods of authorship attribution based on the Markov chain approach are still in use.

To sum up, each method of document identification can be viewed as an "authorship attribution" method; however, the methods based on term similarities should be rather regarded as document classification methods.

3.2 Text Representation

Text representation is supposed to facilitate efficient data processing. A significant variety of methods and data processing models have been developed for this purpose. The common term used to refer to all these techniques is *data mining*, which, in the context of this book, means:

- Extraction and convenient representation of the non-structured information implied in the data;
- Processing of a great body of data for the purpose of pattern recognition;
- Detection of new significant correlations and tendencies in large bodies of data;

- Automatic extraction of efficient information out of large datasets;
- Analysis of the information in a database with the aim of detecting anomalies and trends without decoding the record meaning.

As a rule, all these different models are based on the representation of a text as a set of sensible and formal words, extracted from the text. The first model built on the basis of this approach was the Bag-of-Word (BoW) model.

3.2.1 Bag-of-Words Model

The *The Bag-of-Words (BoW) model* is a popular method for representing documents, which is based on the idea that *"the frequency of word occurrence in an article furnishes a useful measurement of word significance"* (H.P. Luhn, The automatic creation of literature abstracts, IBM J Res Dev 2 (1958), pp. 159165). Although the differences in word frequencies are connected with the content of a particular document, they all obey to the well-known Zipfs law, which will be discussed in detail in Chapter 5.

The BoW model disregards the grammar and even the word order: for example, two semantically different phrases - *She is a pretty girl, isnt she?* and *She isnt pretty, is she a girl?* - are considered as the same text. Thus, in this model, each document looks like a "bag" which contains some words from the dictionary. In the simplest case, only the presence/absence of particular words in the document is considered, the resulting model being called "binary". In a sophisticated version of the BoW model, not only the presence of the words, but also each words weight is taken into account. In this way, the simple model, which considers a set of words, is transformed into a model, which considers a set of word-weight pairs.

In the case of human-language documents, the whole dictionary of the language appears to be excessive for processing a particular text, so the size of the set of words in the dictionary, the so-called *feature space*, should be reduced appropriately. The BoW model usually employs those meaningful words of the language, which are characteristic of the document. H. P. Luhn proposed the Keyword-in-Context (KWIC) indexing technique to discriminate between keywords and non-keyword terms, which he referred to as *stop words.* There is no universal list of stop words since such a list must be language-oriented. For example, the word *the* is one of the first candidates to be included in the list of English stop words, while in French it is quite meaningful (being translated as *tea*). The University of Glasgow published a list of 319 English stop words, e.g. *a*, *an*, *above*, *you*, *yet*. Dividing the words into meaningful and stop words should be also based on a deep expertise in the corpus comprehension. For example, in the above-mentioned phrase - *She is a pretty girl, isnt she?* - only the words *girl* and *pretty* should be regarded as meaningful.

Another possible way to reduce the feature space is to use *stemming*. This term is related to the fact that almost every word has many morphological forms. Consider the following examples.

Example 1. The phrases *She isnt pretty* and *She is not pretty* have different number of words, but they are, obviously, the same phrases since the two forms - *isn't* and *is not* - should be considered as one form.

Example 2. Such words as *computing*, *computable*, *computation*, *compute*, *computes*, and *computed* have the same linguistic root. Treating them as one word in calculations of word occurrences improves document classifications in almost all cases.

One can see, thus, that preprocessing in the BoW model is a complicated task. Another, closely related model, sometimes called *Bag-of-Tokens*, is free from this drawback. Instead of meaningful words, this model, discussed in detail below, employs strings of letters, which neednt have any definite sense (the so-called N-grams).

3.2.2 N-Gram Technique

Another technique which provides embedding a text document into a ‘”bag of words” is the *N-gram* technique. An *N-gram* is a subsequence of N items from a given sequence. N-grams are widely used in statistical natural language processing. N-grams containing 1, 2, 3, 4, or N characters are referred to as a *unigram*, a *bigram* (or a *digram*), a *trigram*, and an *N-gram*, respectively. Perhaps, N-grams were first applied by Shannon [233] for comparison of texts (it was also Shannon who introduced the term N-gram). He considered a “discrete source generating the message, symbol by symbol”. According to this concept, the source could be a text written in a natural language, a continuous information source (e.g., speech), represented in a discrete manner, or an abstract stochastic process which produces a sequence of symbols. Shannon's N-grams were viewed as formal words, not related to their real meaning. Thus, if an object is described as a sequence over a given alphabet, feature extraction can be performed in terms of its subsequences. Shannon was interested in predicting the next-symbol expectancy in a sequence, provided the previous N letters were known. Formally speaking, the N-gram model is intended to predict the identity of the next letter, say x_i, based on $\{x_{i-n}\}, n = 1, ..., N$, i.e., to calculate the conditional probability $P(x_i|x_{i-1}, x_{i-2}, ..., x_{i-N})$. According to the independence assumptions, characteristic of language modeling, a word depends only on the last N characters in the alphabet. Such an assumption simplifies the learning problem for a language model.

Depending on the domain of interest, the current methodology can employ N-grams of different lengths. The effect of N-gram lengths is the subject of considerable discussion in the literature. For instance, the Stores system [102] uses the value of $N = 3$ because it yields the best selectivity in the search access rate. In some other systems, trigrams were also used to save memory or disk accesses [3], while Cavnar [38] employed both bigrams and trigrams. It was suggested that bigrams provided better matching for individual words and trigrams gave better connections between words to recover phrase matching. Cohen [47] and Damashek [54] used 5-grams, while Robertson and Willett [212] used bigrams and trigrams (the choices not being justified in any way). It is also possible to use initially N-grams and then $(N-1)$-grams to improve the results. In this way, Huffman and Damashek [114] and Huffman [113] achieved a nearly 20% improvement of a corrupted text. Bigrams were applied in combination with N-grams for Korean text retrieval, providing the best 11-point average precision [165].

3.3 Geometrical Approach

3.3.1 Vector Space Modeling

The *Vector Space model (VSM)*, also called the *Term Vector model*), was introduced by Salton and his colleagues [223] as further sophistication of the BoW and the N-gram models. It is the most commonly used approach to text document classification. In the Vector Space model, any document is represented as a point in an M-dimensional space, where M is the number of items (the size of the term vocabulary). Thus, an algebraic model is obtained, which embeds text documents into a coordinate space. For this purpose, a set of terms should be selected, the definition of the term depending on the application. Typically, the terms are single words, keywords, or longer phrases and each coordinate corresponds to a separate term. If a term occurs in the document, the value of the corresponding coordinate is non-zero.

To give a more accurate description of the model, consider the document set $\mathbf{D}$, which is represented by a *term-document matrix* A, where each column stands for a document and each item $a_{ij} \in A$ stands for the weighted frequency of term i in the document j::

$$A = [t \times \mathbf{D}] \rightarrow \begin{bmatrix} a_{11} & ... \; a_{1N} \\ ... & ... \; ... \\ a_{M1} & ... \; a_{MN} \end{bmatrix},$$

where $N = |\mathbf{D}|$ is the corpus size and M is the total number of terms. A row of the matrix corresponds to a term furnishing the term relation to the documents:

$$t'_i = \begin{bmatrix} a_{i1} \; ... \; a_{iN} \end{bmatrix},$$

while a column of the matrix,

$$d_j = \begin{bmatrix} a_{1j} \; ... \; a_{Mj} \end{bmatrix},$$

corresponds to a document, furnishing the relation of the document to the terms. The suite of the terms is predefined, e.g., it may be the set of all unique words occurring in the document group. *Boolean Vector models* consider document vectors to have merely zeros and ones as coordinates. Zero corresponds to the absence of the attribute and one appears if the attribute is present. The widespread model fine-tuning is performed through term weighting, which takes into account the occurrences of the attributes (such as keywords and key phrases), the frequencies of all the attributes in the document collection, and their positions in the text (e.g., in the title, header, abstract, or text body). Commonly, the following matrix is considered:

$$\widetilde{A} = \left\{ \widetilde{a}_{ij} = \frac{a_{ij}}{\sum\limits_{l=1}^{M} a_{lj}^2},\ i = 1, ..., M,\ j = 1, ..., N \right\}.$$

Indeed, the matrix $C = \widetilde{A} * \widetilde{A}'$, with the elements

$$C = \left\{ c_{ij} = \sum_{l=1}^{N} \widetilde{a}_{il} \widetilde{a}_{jl},\ i, j = 1, ..., M \right\},$$

contains the correlations between the terms within the documents. In the same way, the matrix $\widetilde{A}' * \widetilde{A}$ contains the correlations between the documents over the terms. For instance, if two terms, i and j, occur in all documents with the same frequencies, $c_{ij} = 1$.

The success of the above representation largely depends on the model of term weighting (the values of the vector coordinates). In the simplest model of such type, the term weight refers to the weighted word frequency in the document. However, the latter value does not consider the frequency of the term occurrence in all the documents of the sample, i.e., the discriminating ability of the term. To eliminate this drawback, it was proposed to use the so-called $tf - idf$ (term frequency - inverse document frequency) weighting, where the weight of word i in document j is proportional to the number of the word occurrences in the document and inversely proportional to the number of the documents (in the sample) in which the word occurs at least once. This value provides a statistical measure for estimating the word significance in a document and in a text corpus. The term significance increases with the increase of its frequency in the document, but it must be related to the total word occurrence (which tends to decrease the significance). Modifications of the $tf - idf$ weighting method are usually used by search engines. The value of $tf - idf$ is calculated by multiplying two expressions. The first expression equals the relative frequency of the term i in the document j, namely,

$$tf_{ij} = \frac{a_{ij}}{\sum\limits_{l=1}^{M} a_{lj}},$$

where the denominator is the total occurrence of all terms in the document j. The relative frequency is considered in order to prevent a bias towards large documents, which can produce high frequencies of all terms regardless of their actual importance. The second expression emphasizes the general term importance and is evaluated using the inverse document frequency

$$idf_i = \log \frac{N}{t_i(\mathbf{D})},$$

where

$$t_i(\mathbf{D}) = |\{d_j \in \mathbf{D} : t_i \in d_j,\ j = 1, ..., N\}|$$

is the number of documents which contain the term t_i (obviously, a positive value). Finally,

$$tfidf_{ij} = tf_{ij} * idf_i.$$

A high value of $tfidf_{ij}$ is caused by a high frequency of the term (in a particular document) relatively to its frequency in the corpus. Thus, this index has a tendency to sort out common terms. The model does not differentiate among dissimilar contexts, as it is unable to account for different meanings that the keyword may have. On the other hand, the "cost" of this model is not great as compared to the methods based on Latent Semantic Analysis (LSA)(see below), which are usually costly for large dynamic text collections. Moreover, the *tf-idf* approach has the following advantages (which make VSM an attractive method):

- It is capable of providing consequence ranking of documents of varied types (e.g., texts, multilingual texts, images, audios, videos) according to the requirement that the document features be well-defined.
- It can handle documents in different languages.
- The Information Retrieval process based on this approach can be performed automatically .

On the other hand, VSM has the following drawbacks:

1. Big documents are imperfectly characterized since they have poor similarity values.
2. Keyword choice should be precisely consistent with the significant document vocabulary, otherwise, a "false positive match" may occur.
3. The model is non-sensitive to the term order in the document.
4. Documents on similar subjects which consist of different word sets are not recognized as being linked. This situation leads to a "false negative match".

3.3.2 Latent Semantic Analysis

One of the main problems associated with the classification of text documents is the high dimension of the term space. The number of the words in a relatively small sample may be as great as thousand hundreds, which leads to low effectiveness of many standard classification methods. In this context, the problem of reducing dimensionality becomes especially important. One possible solution for solving this problem is provided by the *Latent Semantic Analysis* (*LSA*).

LSA is a theory and technique for extracting context-depending word meanings, performed through statistical processing of large text datasets (which reduces the term-space dimension) [56]. In this method, the basis for the analysis is the matrix of the term occurrences in the documents, A, which is referred to as a *term-document matrix*. It is known that any rectangular matrix A can be factored into a product of three matrixes, $A = U\Delta V$, so that the matrixes U and V consist of orthonormalized columns, while Δ is a diagonal matrix of singular values. The product $U\Delta V$ is referred to as a *singular value decomposition* (*SVD*) of matrix A. Such factorization has a remarkable property: if only k largest singular values are preserved in Δ and only the columns corresponding to these values are preserved in U and V, the product of the resulting matrixes yields the best approximation (in the Frobenius norm) of the original matrix A by a k-rank matrix. It should be noted that matrix A is usually an excessively sparse matrix because it is built on the terms present in all documents. It may be useful to take into account synonyms so that the number of zero elements in the matrix is reduced. Moreover, 3the primary term-document matrix is commonly noisy, which results from the fact that many words can have multiple unrelated meanings. Due to the fact that A is noisy and sparse it may be replaced by a low-rank approximation.

Now let us describe, in a more formal way, the reduction of the term-document matrix according to the LSA method.

The LSA approach considers the SVD of matrix A,

$$A = U\Delta V',$$

where

- U is an M-by-M unitary matrix containing the eigenvectors of $AA^{'}$;
- V is an N-by-N unitary matrix containing the eigenvectors of $A'A$;
- Δ is a diagonal matrix with monotonously decreasing diagonal elements δ_i

(see, for example, [108]).

A truncated (approximated) version of the SVD method , which results from the biggest singular values of Δ, is given by

$$A_k = U_k \Delta_k V_k'.$$

This expression is based merely on the column vectors of U and the row vectors of V', the rest of matrix A being disregarded. In the LSA method, this transformation is considered as translating term-document vectors into a concept space. Thus, the coordinates of each vector represent the occurrences of the term in the concepts. Truncated representations of term t_i and document d_j are designated as $\widehat{t_i}$ and $\widehat{d_j}$, respectively. After projecting the documents into the concept space, they can be compared with each other using a vector similarity function. The most common measure for this comparison is the *cosine similarity*, which is defined as

$$\cos(\widehat{d_1}, \widehat{d_2}) = \frac{\left\langle \widehat{d_1}, \widehat{d_2} \right\rangle}{\left|\widehat{d_1}\right| \left|\widehat{d_2}\right|}. \tag{3.1}$$

In summary, LSA is carried out in two steps - projecting and matching. In the first step, the documents are translated by the matrix U_k into pseudo-documents in the concept space. The result is weighted by the corresponding inverse singular values, which means that, for a vector q, the translation must be performed before q is compared with the document vectors in the concept space:

$$\widehat{q_k} = \Delta_k^{-1} U_k q. \tag{3.2}$$

In the second step, similarities between the pseudo-document $\widehat{q_k}$ and the documents in the concept space are calculated using a similarity measure.

The LSA technique [163] is intended to examine the relationships between a document set and document terms through the construction of a concept collection. Establishing such relationships is often *Latent Semantic Indexing* (*LSI*).

3.4 Text Classification Problem

Text classification is usually aimed at identifying the main topic of the text or attributing the document to a certain field of knowledge. First, let us introduce the general notions of the *classification problem*: given a set of classes, C=$\{C_1, C_2, \ldots\ Cj\}$, we seek to determine which class (or classes) a given object, $d \in X$, belongs to (d is the description of the document and X is the *document space.* Classes are also called *categories* or *labels.*

The classification problem could be solved in the framework of the Bag-of-Words model, using the Bayes approach. However, we will consider the problem for data organized in terms of VSM. In this case, the theory of linear and non-linear classificators can be applied.

3.4.1 Linear Classification

In the *linear classification* approach, each text object is represented by a vector in an n-dimensional space $x_{i,}$ $i = 1, \ldots, p$. In the simplest case, each point in the vector space belongs only to one of the two classes, say, to the first or to the second class. The typical problem of linear discrimination is whether the points can be separated by a hyper-plane $wx = d$, where w is a fixed vector and x is the corresponding variable. Since such hyper-planes may be quite numerous, it would be reasonable to assume that the maximization of the gap between the classes facilitates the classification. In other words, we must find a hyper-plane located at the maximum distance from the nearest point. Such hyper-plane is called the *optimal discriminating hyper-plane*, while the corresponding linear classification is called the optimal discriminating classificator. The problem of finding the optimal discriminating hyper-plane can be reduced to minimizing the length of a certain vector w under the condition $(wx_i) - d \geq 1$ if x_i belongs to the first class or under the condition $(wx_i) - d \leq 1$ if x_i belongs to the second class. The choice of the condition depends on which of the two sets to be discriminated the point x_i belongs to. Let ϵ_i equal 1 if the item with the index i belongs to one particular set of the two sets under consideration and -1 if the same item belongs to the other set. Thus, we have a square optimization problem

$||\mathrm{w}||^2 \rightarrow min,$

$\epsilon_i((wx_i) - d) \geq 1,\ 1 \leq i \leq p.$

3.4.2 Non-linear Classification and Kernel Trick

The method of linear classification can be extended to the case of non-linear separability. The *non-linear classificator* was designed on the basis of the so-called *kernel trick*, where standard scalar products are substituted by arbitrary kernels, which allow to build non-linear discriminators. The resulting algorithm is similar to that of linear classification, the only difference being the replacement of each scalar product in the above expressions by a non-linear kernel function (a scalar product in a higher-dimensional space).

Kernel methods are applied in order to simplify the problem by mapping the source space into a certain new space. This procedure can result in reducing a non-linear problem to a linear one. The *kernel* stands for the similarity between two objects (documents, terms, string, N-grams, etc.) and is expressed by means of the dot product in the new vector space. Formally speaking, we attempt to embed a non-empty set (the domain) which the patterns are taken from into a Euclidean higher-dimensional feature space, where suitable distance concepts can be applied. First, the objects are mapped by the function

$$\phi : X \rightarrow F; x \rightarrow \phi(x) \tag{3.3}$$

and, subsequently, are evaluated by means of the dot product

$$K(x, y) = \langle \phi(x), \phi(y) \rangle . \tag{3.4}$$

The dimensionality of the feature space F is typically much higher than that of the input space X. However, the dimensionality of a manifold inhabited by expanded input vectors cannot be higher than that of the input space. For instance, if the input space is two-dimensional and the feature space is three-dimensional, the two-dimensional manifold of the input space is embedded in the three-dimensional feature space. For high-dimensional feature spaces, the mapping into and the computations within the feature space can result in a computationally difficult task. On the other hand, by means of mapping, various inner products, in particular, feature spaces, can be efficiently obtained. It is important to note that the feature space and its inner product needn't be taken into consideration because all calculations are performed using the kernel function. The only requirement that the kernel function must meet is the existence of an inner product in the consequent feature space. In what follows, we present several well-known kernels.

3.4.2.1 Polynomial Kernels

Mapping from R^2 into the feature space R^6 is defined as

$$\begin{aligned}
&\phi_1 = 1;\\
&\phi_2 = 2\sqrt{x_1};\ \phi_3 = 2\sqrt{x_2};\\
&\phi_4 = x_1^2;\ \phi_5 = \sqrt{2}x_1 x_2;\ \phi_6 = x_2^2.
\end{aligned}$$

Calculating the Euclidian inner product of the input vectors mapped into the feature space, one obtains

$$\begin{aligned}
&K(x, y) = \phi^T(x)\phi(y) = 1 + 2x_1y_1 - 2x_2y_2 + x_1^2y_1^2 - 2x_1y_1x_2y_2 + x_2^2y_2^2 =\\
&= (1 + x^T y)^2.
\end{aligned}$$

This mapping can be generalized to higher exponents in the following way:

$$K(x, y) = \phi^T(x)\phi(y) = (1 + x^T y)^n.$$

3.4.2.2 Radial Basis Function (RBF)

In this case, the kernel has the form

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma^2}\right),\ \sigma > 0.$$

The geometrical interpretation of the polynomial and the RBF kernels is presented in Fig. 3.1.

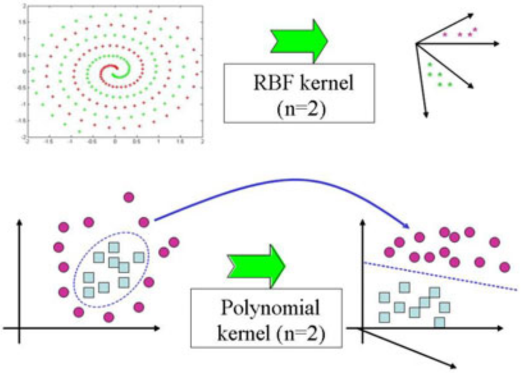


Fig. 3.1 The geometrical interpretation of the polynomial and the RBF kernels

3.4.2.3 Sigmoid Kernel

Sigmoid kernel can be represented by

$$K(x, y) = \tanh(\gamma x^T y + c). \tag{3.5}$$

For $\gamma > 0$, γ and c can viewed as the scaling and shifting parameter of the input data, respectively. If $\gamma < 0$, the dot-product is reversed.

Let K be a real symmetric function, i.e., $K(x, y) = K(y, x)$ for any $x, y \in \mathbf{X}$. The function K is a positive definite kernel if, for any $s \geq 1$ and any $x_1, .., x_s \in \mathbf{X}$, the matrix defined by $K_{ij} = K(x_i, x_j)$ is positive definite, i.e.,

$$\sum_{i,j=1}^{s} K_{ij} c_i c_j \geq 0, \tag{3.6}$$

for any real $c_1, .., c_s$. If the equality is reached only at $c_i = 0$, $i = 1, .., s$, the kernel is called *strongly positive definite*. It is well-known that (3.6) holds if and only if K is a Mercer kernel [187]. A real symmetric function $\mathfrak{S}$ is negative definite if, for any $s \geq 1$ and any $x_1, .., x_s \in \mathbf{X}$,

$$\sum_{i,j=1}^{s} \mathfrak{S}(x_i, x_j) c_i c_j \leq 0$$

for any real $c_1, .., c_s$ such that

$$\sum_{i=1}^{s} c_i = 0.$$

The kernel is called *strongly negative definite* if the equality is reached only at $c_i = 0, \quad i = 1, .., s$. Obviously, if K is positive definite, $(-K)$ is negative definite, but the opposite does not generally hold. Note that the kernel $(-\mathfrak{S})$ is often called a *conditionally positive definite* kernel. The connection between the so-called *infinitely divisible* kernels and the negative definite kernels has been discussed in the literature (see, for example [101]). A similar connection is well known in calculus and in the probability theory (see, for example [7] and [173]). A generalization of this relationship was considered in [273].

If f is the characteristic function of a symmetric distribution Q on $\mathbf{R}^1$, then the function

$$K(x_i, x_j) = f(x_i - x_j)$$

is a positive definite kernel. Moreover, if Q is an infinitely divisible distribution (see, for example [173], part 11), f can be written as

$$\phi(x) = -\log(f(x)) = -\int \frac{(\cos(tx) - 1)}{t^2} \nu(dt),$$

where ν is a finite symmetric measure on $\mathbf{R}^1$ and the kernel

$$\mathfrak{S}(x_i, x_j) = \phi(x_i - x_j)$$

is negative definite (see, for example [177]). In particular, functions of the type

$$\phi(x) = \|x\|^r, \ 0 < r \leq 2$$

produce negative definite kernels, which are strongly negative definite if $0 < r < 2$. The case of $r = 2$ corresponds to the Gaussian law.

It is important to note that a negative definite kernel, $\mathfrak{S}_2$, can be obtained from another negative definite kernel, $\mathfrak{S}_1$, through certain transformations, for example: $N_2 = \mathfrak{S}_1^\alpha, \quad 0 < \alpha < 1$ and $\mathfrak{S}_2 = ln(1 - \mathfrak{S})$ [227].

3.4.3 Kernel N-Gram Techniques

A group of kernel functions which are defined on feature vectors obtained from strings is called *string kernels* [280], [101]. These kernels are based on the features derived from the occurrences of specified string subsequences. There exist *contiguous* and *non-contiguous* kernels, which have bounded and unbounded lengths, respectively.

3.4.3.1 *N*-Spectrum Kernels

It has been shown above that each string over an alphabet $\mathfrak{A}$ can be described by its possible contiguous substrings of length N, i.e., by means of N-grams. Each sequence of letters defines a certain frequency distribution on a given N-gram set, each frequency corresponding to a particular N-gram occurrence in the sequence S. This distribution can be considered as a vector of a certain vector space. As a result, each string S over the alphabet $\mathfrak{A}$ is mapped into an N-gram feature space $\mathfrak{A}^{(N)}$ through the feature map

$$\Phi(S) = \{\phi_u(S)\} = \left\{\text{number of occurrences of } u \in \mathfrak{A}^{(N)} \text{ in } S\right\}.$$

This conversion makes it possible to compare the sequences in an efficient way and consider each string as an item depicted by its coordinates indexed in the N-gram space $\mathfrak{A}^{(N)}$. The similarity between two strings, S_1 and S_2, is defined by

$$K(S_1, S_2) = \langle \phi_u(S_1), \phi_u(S_2) \rangle .$$

Another kernel version is created by assigning binary values, 0 or 1, to the coordinates depending on the presence of the appropriate N-gram.

In the N-gram approach, the feature space has a very high dimension even for fairly small values of N. At the same time, the feature vectors are sparse because, for each string S, the number of non-zero coordinates is bounded by the value $length(S) - N + 1$ (which equals the total number of N-grams in the sequence S). This property makes it possible to efficiently calculate kernel values. The suffix tree for N-grams occurring in S_1 and S_2, which is obtained by moving an N-length sliding window across S_1 and S_2, is a powerful tool facilitating the determination of the similarity between S_1 and S_2. The suffix tree can be created in time $O(N * length(S))$, which is linear both in S and N, according to the algorithm proposed in [263]. It is possible to build a suffix tree for all considered sequences simultaneously and handle all the kernel values in one traversal of the tree.

Let us consider the following simple example. Let the alphabet $\mathfrak{A}$ consist of four letters: $\mathfrak{A} = \{A, T, C, G\}$. The short sequences $S_1 = ACCGGC$ and $S_2 = AACCTGGC$ are compared using 3-grams ($N = 3$). These sequences are mapped onto the feature spaces

$$\Phi(S_1) = \{ACC, CCG, CGG, GGC\}$$

and

$$\Phi(S_2) = \{AAC, ACC, CCT, CTG, TGG, GGC\}.$$

In view of the fact that the 3-grams AAC and GGC appear in both sequences, we get

$$K(S_1, S_2) = 2.$$

Applying the above transformation, we lose, evidently, certain information about the strings. Generally speaking, this technique may lead to a complete loss of information. For example, given the alphabet $\{a, b, c\}$, both strings "bcacb" and "cacbc" produce exactly the same 2-grams.

3.4.3.2 *N*-Weighted Subsequence Kernels

Feature space can be indexed by all elements of $\mathfrak{A}^{(N)}$. For a given sequence $S = \{s_i, \quad i = 1, ..., N\}$, we introduce an increasing sequence of indexes $\mathbf{i} = \{i_t, t = 1, \ldots, m, m \leq N\}$ and designate

$$s(\mathbf{i}) = \{s_{i_m}, i \in \mathbf{i}\} \in \mathfrak{A}^{(m)}.$$

Note that this sequence is not necessarily contiguous. For instance, if $S = ABCDE$ and $\mathbf{i} = [1, 3, 5]$, $S(\mathbf{i}) = ACE$. If we denote by $l(\mathbf{i})$ the length generated by $S(\mathbf{i})$, then

$$l(\mathbf{i}) = i_m - i_1 + 1.$$

It can be seen that the length $\mathrm{l}(i)$ includes both matching symbols and gaps. The *gap-weighted feature map* is defined as the sum of gap weights of the N-gram occurrences, u, in a (non-contiguous) subsequence of S. If the gap penalty is denoted by λ and the weight of the length penalty is $\lambda^{l(u)}$, we obtain the following expression for the gap-weighted feature map:

$$\Phi(S) = \{\phi_u(S)\} = \left\{_{\mathbf{i}:u=S(\mathbf{i})}\lambda^{l(u)}\right\}.$$

All possible subsequences of N symbols are being matched to one another even if these subsequences are not consecutive and each considered subsequence is “discounted” by its total length. The decay factor is applied to the gaps and to the matching symbols. So, if λ equals 1, the gaps are not taken into account when the value of the feature is calculated. If λ is 0.5, each gap symbol results in dividing the feature value by 2. To keep the kernel values comparable for various values of N and to make them independent of the string length, the normalized version of the feature map can be used:

$$\widehat{K}(S_1, S_2) = \frac{K(S_1, S_2)}{\sqrt{K(S_1, S_1)K(S_2, S_2)}}.$$

The normalized feature map is, actually, an extension of the “cosine normalization” (3.1).

Example 1. Let

- $\mathfrak{A} = \{A, T, C, G\}$;
- $S_1 \quad : \quad TTCGGAGAGTGTG$;
- $S_2 \quad : \quad CTCTATCG.$

It is easy to see that $S_{1,CAT} = 2\left(\lambda^8 + \lambda^{10}\right)$ and $S_{2,CAT} = \quad \lambda^4$. Hence ,

$$K(S_1, S_2)_{CAT} = 2\lambda^4\left(\lambda^8 + \lambda^{10}\right).$$

Example 2. This example is taken from [36]. Let

- $\mathfrak{A} = \{A, T, C, G\}$;

- S_1 : $CATG$;
- S_2 : $ACATT$.

For $N = 3$, we obtain:

u	$\phi_u(S_1)$	$\phi_u(S_2)$	u	$\phi_u(S_1)$	$\phi_u(S_2)$
AAT	0	$\lambda^4 + \lambda^3$	CAG	λ^4	0
ACA	0	λ^3	CAT	λ^3	$\lambda^3 + \lambda^4$
ACT	0	$\lambda^4 + \lambda^5$	CTG	λ^4	0
ATG	λ^3	0	CTT	0	λ^4
ATT	0	$\lambda^3 + \lambda^5$	$others$	0	0

The fact that the only feature for which both sequences have non-zero values is CAT leads to the following expression:

$$K(S_1, S_2) = \lambda^3 \left(\lambda^3 + \lambda^4\right).$$

On the one hand, direct calculation of the above-mentioned kernels cannot be performed even for small values of N; on the other hand, it employs a more efficient recursive dynamic-programming approach [174]. Another efficient algorithm [217] is based on the following reasoning. Let us suppose that the value of the kernel for two strings S_1 and S_2 is already known. How can we calculate $K(S_1 \cup \{x\}, S_2)$ for a particular $x \in \mathfrak{A}$? It is easy to see that

- All subsequences common to S_1 and S_2 are also common to $S_1 \cup \{x\}$ and to S_2.
- All new matching subsequences ending in x which are present in S_1 and whose $(N-1)$-symbol prefix is found in S_1 (possibly non-contiguously) must be taken into account.

The proposed recursive implementation can be summarized in the following way:

- $K_0^{'}(S_1, S_2) = 1$ for all S_1 and S_2;
- $K_i^{'}(S_1, S_2) = 0$ if $min(length(S_1), length(S_2)) < i \quad (i = 1, ..., N-1)$;
- $K_N(S_1, S_2) = 0$ if $min(length(S_1), length(S_2)) < N$;
- $K_i^{'}(S_1 \cup \{x\}, S_2) = \lambda K_i^{'}(S_1, S_2) +_{j:S_2,j=x} K_{i-1}^{'}(S_1, S_2[1:j-1])$ $\lambda^{length(S_2)-j+2}\,(i = 1, ..., N-1)$;
- $K_N(S_1 \cup \{x\}, S_2) = K_N(S_1, S_2) +_{j:S_2,j=x} K_{N-1}^{'}(S_1, S_2[1:j-1])\lambda^2$,

where $S_2[1:j-1]$ is a substring of S_2 composed of the items in the positions from 1 to $j-1$ and $S_2[j=x]$ is a subsequence of S_2 in which x is the j^{th} element. The complexity of this computation is $O(N * length(S_1) * length(S_2)^2)$. In the case of Example 2, we obtain:

u	S_1	S_2	u	S_1	S_2
AA	0	λ^5	GA	0	0
AC	0	λ^5	GC	0	0
AG	λ^3	0	GG	0	0
AT	λ^3	$2(\lambda^3+\lambda^5)$	GT	0	0
CA	λ^4	λ^4	TA	0	0
CC	0	0	TC	0	0
CG	λ^4	0	TG	λ^2	0
CT	λ^4	$2\lambda^4$	TT	0	λ^2

Note that the feature $u = CT$ leads to the value $2\lambda^4$ for the string $S_2 = ACATT$. This results from the fact that the two occurrences of u begin in the second symbol C which is located at the distance of four symbols from the end of t. Thus,

$$K_2^{'}(CATG, ACATT) = 2\lambda^6 + 5\lambda^8.$$

3.4.4 Euclidean Embeddings

Euclidean embeddings, similar to Lipschitz [105] and Bourgain [27] embeddings (the latter being an important special type of the former), offer an alternative tool for mapping of non-Euclidean metric spaces into Euclidean spaces. In this way, each object is associated with a Euclidean vector, such that the distances between the items are related to Euclidean distances between the object images. Euclidean embeddings provide a means for decreasing extensive retrieval time when the estimation of dissimilarities appears to be computationally "expensive" in the source space.

Let us consider a distance function D_X defined on a space $\mathbf{X}$:

$$D_X : \mathbf{X} \times \mathbf{X} \to \mathbf{R}_+ .$$

Obviously, the following extension can be performed: if R is a subset of $\mathbf{X}$, then

$$D_X\,(x, R) = \min_{r \in R}\{D_X\,(x, r)\}$$

is the distance from $x \in \mathbf{X}$ to its closest neighbour in R. Given a subset R, a simple one-dimensional Euclidean embedding is defined as

$$F^R(x) = D_X(x; R).$$

The set R is called a *reference set.* If R consists of a single object r, this object is usually called a *reference* or a *vantage object* [105]. If D_X satisfies the triangle inequality, F^R transforms close points in $\mathbf{X}$ into close points on the real line. Quite often, D_X disobeys the triangle inequality for some object triples. However, F^R can still transform close points in $\mathbf{X}$ into close points in R, at least, in most of the cases [14]. Unfortunately, far-away items can also be transformed onto nearby points, while multidimensional embedding is unlikely to produce such an effect. It is possible to select n different reference sets $\{R_1, R_2, ..., R_n\}$ and consider the embedding

$$F(x) = \left\{F^{R_1}(x), F^{R_2}(x), ..., F^{R_n}(x)\right\},$$

which is referred to as *Lipschitz embeddings*. Transformations of this kind are, obviously, highly non-linear and can hardly be intuitively interpreted. However, some information can be obtained if the points of $\mathbf{X}$ are close to each other and constitute well-separated clusters. In this case, if a point of a reference set R belongs to cluster $\mathfrak{C}_1$ and does not belong to cluster $\mathfrak{C}_2$, $D_X\left(x, R\right)$ is relatively small for $x \in \mathfrak{C}_1$ and relatively large for $x \in \mathfrak{C}_2$. Thus, the coordinate induced by R accounts for large distances between the points belonging to $\mathfrak{C}_1$ and those belonging to $\mathfrak{C}_2$. Obviously, we cannot consider the set $\mathbf{X}$ as a union of well-separated sets arranged like a distinct cluster pattern. Bourgain [27] demonstrated that there exists a reference set collection which is suitable for general inputs. According to this approach, reference sets are selected and the corresponding embedding is performed by means of a randomized procedure. The suggested reference collection R includes $O(log_2 \left|\mathbf{X}\right|)$ sets X_{ij}, which can be represented as a table organized into columns and rows in the following way:

$$\left.\begin{array}{||c||c||c||c||}
\hline
X_{11} = \{x_{_{11,1}}, x_{_{11,2}}\} & X_{12} = \{x_{_{12,1}}, x_{_{12,2}}\} & ... & X_{1k} = \{x_{_{1k,1}}, x_{_{1k,2}}\} \\
\hline
X_{21} = \{x_{_{21,1}}, ..., x_{_{21,4}}\} & X_{22} = \{x_{_{22,1}}, ..., x_{_{22,4}}\} & ... & X_{2k} = \{x_{_{2k,1}}, ..., x_{_{2k,4}}\} \\
\hline
.... & .... & ... & .... \\
\hline
X_{\beta 1} = \{x_{_{\beta 1,1}},, ..., x_{_{\beta 1,2^\beta}}\} & X_{\beta 2} = \{x_{_{\beta 2,1}}, ..., x_{_{\beta 2,2^\beta}}\} & ... & X_{\beta k} = \{x_{_{\beta k,1}}, ..., x_{_{\beta k,2^\beta}}\} \\
\hline
\end{array}\right\}\beta$$

Here, $k = \beta = O(\log_2 \left|X\right|)$.

In the above table, each set X_{ij} is a random subset of $\mathbf{X}$ of size $2^i, \quad j = 1, ..., k$. Thus, we obtain reference sets of sizes 2, 4, etc., up to the size of approximately $\left|\mathbf{X}\right|$ and the Bourgain embedding peaks at $O(log_2^2 \left|\mathbf{X}\right|)$ dimensions. Bourgain embeddings have the advantage of minimizing the maximum stretch of preset distances (the embedding distortion). Namely, if R is selected as described above, the corresponding embedding distortion is $O(log \left|\mathbf{X}\right|)$. Indeed,

$$D(x_i, x_j) \leq C \log_2 \left|\mathbf{X}\right| * \left\|F^R(x_i) - F^R(x_j)\right\|$$

and

$$D(x_i, x_j) \leq \frac{\left\|F^R(x_i) - F^R(x_j)\right\|}{C \log_2 \left|\mathbf{X}\right|}$$

for each pair (x_i, x_j) and a certain constant $C > 0$, $\|*\|$ being an Euclidian metrics. The $\log_2 \left|\mathbf{X}\right|$ bound is tight in the sense that there exist spaces $\mathbf{X}$ where smaller distortions cannot be achieved.

Let us consider an example of the Bourgain embedding in which $|\mathbf{X}| = \{x_1, x_2, ..., x_8\}$ and the distances between the points are specified as

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	10	9	8	7	6	5	4
x_2	10	0	8	12	10	10	10	8
x_3	9	8	0	10	12	8	9	5
x_4	8	12	10	0	5	8	11	12
x_5	7	10	12	5	0	10	11	12
x_6	6	10	8	8	10	0	8	9
x_7	5	10	9	11	11	8	0	11
x_8	4	8	5	12	12	9	11	0

According to the Bourgain approach, 6 reference sets must be created at random in such a way that 3 sets have 2 elements each and the other 3 sets have 4 elements each. For instance, one can select

x_1, x_2	x_3, x_4	x_5, x_8
x_2, x_4, x_6, x_7	x_1, x_5, x_7, x_8	x_3, x_4, x_5, x_6

As an example, let us compute the embedding of x_1 and x_2:

$$F^R(x_1) = \begin{array}{|c|c|c|}\hline 0 & 8 & 4 \\ \hline 5 & 0 & 6 \\ \hline \end{array}, \ F^R(x_2) = \begin{array}{|c|c|c|}\hline 0 & 8 & 8 \\ \hline 0 & 8 & 8 \\ \hline \cdot & & \end{array}$$

It follows from the above that

$$\left\| F^R(x_1) - F^R(x_2) \right\| \approx 10.44.$$

The original distance between x_1 and x_2 is 10. Bourgain embeddings have some serious drawbacks, namely:

- The number of dimensions, $O(\log_2^2 |\mathbf{X}|)$, produced by the embeddings is so large that the computations become very "expensive". For instance, a set $\mathbf{X}$ of size 1024 links to 100 dimensions.
- It is highly probable that each item of $\mathbf{X}$ can be selected in some reference set; as a result, $\binom{|X|}{2}$ distances must be calculated in order to construct the embedding.

A heuristic embedding, SparseMap simplification [110], decreases the embedding time by computing only $O(\log_2^2 |\mathbf{X}|)$ distances for each entity. Another way of enhancing the embedding efficiency is using a relatively small random subset $\mathbf{X}' \subset \mathbf{X}$ and choosing $O(\log_2^2 \left|\mathbf{X}'\right|)$ reference subsets. Thus, any item is mapped by calculating its distances from each object of $\mathbf{X}'$ [13]. In this case, the embedding is optimal only for the objects belonging to $\mathbf{X}'$ and may not be optimal from the point of view of distortion in the entire set $\mathbf{X}$. Although the Bourgain embedding may be optimal, it is not necessarily superior to other methods. However, the worst-case bound on the distortion is not very tight and is not realized in actual applications.

Chapter 4
DNA Texts

4.1 DNA Information: Metaphor or *Modus Operandi?*

Throughout the history of mankind, its thinkers have speculated that, within the sperm or egg, there exists a "draft" which ensures the development of a frog from a frog egg and a human beeing as an offspring of humans. It may have been Erwin Schrodinger who first coined the term "code" in the context of heredity in his extremely influential book "What is life?" [228]. He wrote, *"It is these chromosomes, or probably only an axial skeleton fibre of what we actually see under the microscope as the chromosome, that contain in some kind of code-script the entire pattern of the individual's future development and of its functioning in the mature state. Every complete set of chromosomes contains the full code; so there are, as a rule, two copies of the latter in the fertilized egg cell, which forms the earliest stage of the future individual. In calling the structure of the chromosome fibres a code-script we mean that the all-penetrating mind ... could tell from their structure whether the egg would develop, under suitable conditions, into a black cock or into a speckled hen, into a fly or ... a woman. But the term code-script is, of course, too narrow. The chromosome structures are, at the same time, instrumental in bringing about the development they foreshadow. They are law-code and executive power -or, to use another simile, they are architect's plan and builder's craft - in one."* Watson and Crick [281] suggested that *"the precise sequence of the nucleotide bases in the code carries the genetical information."* This hypothesis brought about, in 1950s, the emergence of cybernetics and information theory. This theory formulated the scientific principle which holds that a sequence of a limited assortment of building blocks, represented by letters in a text, can carry one or more messages. From this time on, the fundamental processes of life have been described as information storage, copying, and transfer. Yockey [289] noted that the genetic information system is similar to the algorithmic language used in computers in that this system is segregated, linear, and digital.

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 43–60, 2010.
springerlink.com

In many fields of computational biology, a DNA sequence (otherwise named DNA primary structure) is considered as a plenipotentiary representative of a DNA molecule. Is it the Absolute Truth, almost the Truth or just a metaphor? (This question resembles the title of Mikhail Gelfand's article [89].) On the one hand, the DNA is commonly referred to as an information storage or an information carrier. On the other hand, scientists from various fields doubt the appropriateness of such borrowings in molecular biology, arguing that the genome information content cannot be assessed since the key parameters (e.g., signal, noise, message, channel) cannot be properly quantified [136]. Linguists hold that the DNA text lacks such language features as phoneme, punctuation marks, and intersymbol restrictions. Informaticians do not accept the applicability of entropy-oriented information measures to living entities. Statisticians wonder whether Markov models are applicable to nonrandom sequences of nucleotides, etc.

The discussion of these interesting issues is out of the scope of our book. However, because the main purpose of this section is clarifying our *modus operandi* [1] in dealing with the DNA text, we could not avoid some elaboration on the topic.

Similar to other computational biologists, we consider the DNA sequence to be fully "qualified" to carry all genetic information, be it actually the truth or just a metaphor. A DNA sequence presented as a string over the alphabet A, C, G, T is called a DNA text.

In this book, by "DNA text" we often mean a genome (see Section 1.6) or a fragment of a genome. Namely, a genome is the complete collection of hereditary information "coded" as a certain sequence of DNA nucleotides. The fact that DNA is a linear biopolymer allows us to assume that a linear text serves as a rather adequate presentation of a DNA molecule from the informational point of view. The DNA text is written using the alphabet of four letters A, C, G, T; consequently, a genome is a long text over this alphabet.

There are a lot of different meanings of the term "text". According to one definition, "Text is an arrangement of symbols in groups to express defined and recognized meanings". In the case of information encoded in DNA sequences, we would rather avoid the term "meaning" and replace it by the term "instruction", which is defined for and recognized by different "receivers". A CNC program is an example of a text that is a stream of instructions to be performed by a machine. This stream of symbols is converted into a set of defined instructions for the machine tools to perform certain defined operations. So, we would rather reformulate the general text definition in the following way: "A DNA text is an array of symbols designating DNA basic elements, nucleotides. The symbols arranged in groups provide definite and recognizable instructions." These instructions are destined to rather different "reading devices" (some of them were mentioned in Chapter 1). For example,

[1] *Modus operandi* is a Latin phrase, which means *"method of operating or functioning".*

we may say metaphorically that a ribosome gets an instruction to translate a fragment of a DNA text into a string over the alphabet of 20 amino acids. Actually, a ribosome gets, as an input, a fragment larger than the piece to be translated, and recognizes the symbol where the translation should start (it means that a ribosome reads and interprets the instruction designating the start of translation). Another example is promoter recognition by RNA polymerase, which receives instructions about the location of the transcription starting point distributed in a pretty fuzzy way due to the involvement of quite a few transcription factors (see Chapter 1). Such instructions are sometimes called "biological codes" [257].

4.2 DNA Language: Metaphor or Valid Term?

In this section, we are going to discuss the term "language" together with the previously suggested term "text" with reference to DNA. Indeed, it would be natural to relate any text to the language that the text is written in. However, there is no definition of the term "language" that would be equally applicable to all its modern usages. The reason for this lies in the wide expansion of the number of objects that this term is applied to: language of bees, dolphins' language, programming language, queries language, DNA language, etc. Consequently, one can find quite a few different definitions of the term. *Linguists* define a language "as a system of visual, auditory, or tactile symbols of communication and the rules used to manipulate them". *Mathematicians* define a language "as a set of strings over the given alphabet $\mathfrak{A}$, including an empty string $\varepsilon_\emptyset$". A *formal* language is often defined "as a full set of character strings produced by a combination of formal grammar and semantics of arbitrary complexity". A *programming* language is a formal language that can be used to control the behavior of a machine (for example, a computer) in order to get it perform specific tasks. Programming languages are defined on the basis of syntactic and semantic rules.

Sometimes, we prefer to describe a language without introducing any formal grammar which would generate the strings of the language. For example, pattern grammars[2] present a way of language description. Here, we use "pattern description" in a rather informal way, considering a pattern as a string of characters which describes the way the strings should be generated. For example, denoting by α and β two arbitrary strings over the alphabet $\mathfrak{A}$ ($\alpha, \beta \in \mathfrak{A}^*$), one can define the pattern $P_1 = \alpha Y \hat{Y} \alpha \beta$, where Y is a short string ($Y \in \mathfrak{A}^*$ & $|Y| < K$) and $\hat{Y}$ is a string similar to Y. Two strings are considered similar if the distance between them is less than a certain predefined threshold ε ($Y \in \mathfrak{A}^*$ & $dist(Y, Y) < \varepsilon$).

[2] *Pattern grammar* is a corpus-driven approach to English lexical grammar [115].

4.3 Formal Grammars

A *formal grammar* (**G**) defines (or generates) a *formal language* (**L**), which is a (possibly infinite) set of strings ($\mathbf{L(G)} \subseteq \mathfrak{A}^*$). A formal grammar of this type consists of

- a finite set $\mathfrak{A}$ of *terminal symbols* (an alphabet, which is usually represented by lower-case letters);
- a finite set N of *non-terminal symbols* (usually represented by upper-case letters);
- a finite set P of *production rules*, which determine the rules by which the string are created, namely: $\alpha \to \beta$, where $\alpha, \beta \in (N \cup \mathfrak{A})^*$;
- a *start symbol* S_i;
- $\mathfrak{A} \cap N = \emptyset$.

Thus, the language is a set of strings which is generated by applying production rules to a sequence of symbols which initially contains only the start symbol. A rule may be applied to a sequence of symbols by replacing the symbols on the left-hand side of the rule with those that appear on the right-hand side. A sequence of rule applications is called a *derivation.* Such a grammar defines a formal language of all the words consisting solely of terminal symbols; the language can be constructed by a derivation from the start symbol.

4.3.1 Examples of Formal Languages

Example 1. Let $\mathfrak{A}$ be the set $\{a,b\}$ of terminal symbols, N be the set $\{S,A,B\}$ of non-terminal symbols, S be the start symbol, and ε be the empty string. Seven production rules are introduced:

$$S \to ABS; S \to \varepsilon;$$

$$BA \to AB; BS \to b; Bb \to bb;$$

$$Ab \to ab; Aa \to aa.$$

It can be shown that such a grammar defines the language of all words of the form $a^n b^n$ (i.e., n copies of a followed by n copies of b).

Example 2. Let $\mathfrak{A}$ be the set $\{a,b\}$ of terminal symbols, N be the set $\{S\}$ of non-terminal symbols, and S be the start symbol. Two production rules are introduced:

$$S \to aSb; S \to \varepsilon.$$

Similar to Example 1, such a grammar defines the language of all words of the form $a^n b^n$, but does so in a far more natural and simple way.

Example 3. Let $\mathfrak{A}$ be the set $\{a,b,c\}$ of terminal symbols, N be the set $\{S$,A,B$\}$ of non-terminal symbols, and S be the start symbol. Five production rules are introduced:

$$S \to B, A \to a, A \to ac, B \to b, D \to cb.$$

In such a grammar, there exist only 4 valid derivations:

$$(1) \quad S \to B \to aB \to ab;$$

$$(2) \quad S \to AB \to aB \to acb;$$

$$(3) \quad S \to AB \to acB \to acb;$$

$$(4) \quad S \to AB \to acB \to accb.$$

In this case, the language **L** produced by means of the grammar **G** is

$$\mathbf{L}(\mathbf{G}) = \{ab, acb, accb\}.$$

By the way, there are two derivations for the string *acb*.

4.3.1.1 Examples of Pattern Languages That Are Not Produced by Formal Grammars

Example 4. Let $\mathfrak{A}$ be the set $\{a, b, c, d\}$, N be the set $\{X, Y\}$, X and Y being variables. Let $\boldsymbol{R}$ be the pattern X^2aY^2bXYcd.

The production rule is: replace X and Y by any nonempty strings from $\mathfrak{A}^*$ (all the occurrences of the same variable must be replaced by the same chosen string of terminal symbols).

The following strings belong to the defined language $\boldsymbol{L(R)}$: $ddaccccbdcccd \in$ **L(R)** (replacements: $X \to d; Y \to cc$); $a^7b^3a^3bcd \in$ **L(R)** (replacements: $X \to a^3; Y \to b$), etc.

Example 5. Let $\mathfrak{A}$ be the set {A, G, C, T}, N be the set {X, Y}. The production rules are:

$$X \to \alpha, \alpha \in \mathfrak{A}^*, |\alpha| \geq k_1; Y \to \beta, \beta \in \mathfrak{A}^*, |\beta| \leq k_2.$$

Let pattern **R** be the pattern $XYY^{mir}X^{pal}$, where Y^{mir} is the mirror of Y or, in other words, the string Y rewritten from right to left. Let X^{pal} be a palindrome string obtained from X by rewriting it from right to left with simultaneous complementary substitutions ($A \to U, U \to A, C \to G, G \to C$). In the case of k_1= 6, k_2 = 4, the pattern R produces stem-loop structures with symmetric loops[3] (see Fig. 4.1).

In 1993, a pattern language, PALM, was proposed to describe the patterns of the sequences dealt with in molecular biology [103]. There were a few

[3] **Stem-loop** intramolecular base pairing is a pattern that can occur in a single-stranded DNA or, more commonly, in RNA. The structure is also known as a **hairpin.** It is observed when two regions of the same molecule base-pair to form a double helix that ends in an unpaired loop. The resulting lollipop-shaped structure is the key building block of many RNA secondary structures.

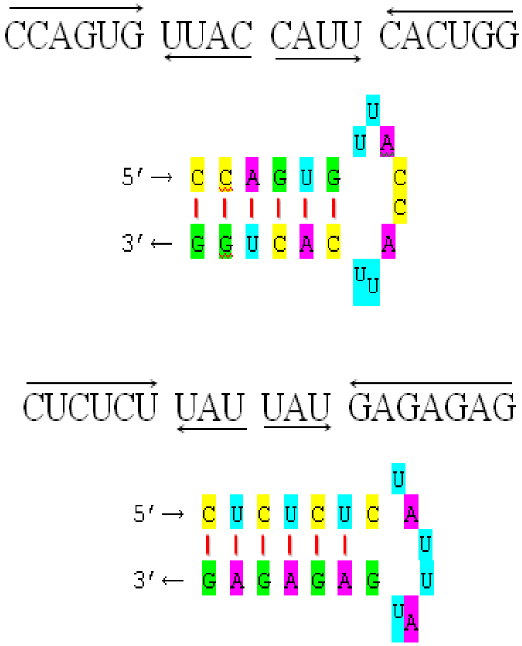


Fig. 4.1 Examples of stem-loop structures with symmetric loops

attempts to invent formal grammars of DNA languages (reviewed by Collado-Vides [49], [50], [51] and by Searls [230], [231], [232]). It seems, though, that the corpus-driven approach is more naturally applicable to the case of DNA languages. This approach is based on corpus compilation, which, in turn, requires understanding of the DNA language words.

4.4 Evolution of DNA Texts

In this section, we discuss the question of how DNA (genetic) texts have originated. The origin of all kinds of texts that appear in human society is obvious and trivial: they are written by human beings. It can be added that computer texts may also be considered as written by "human machines" and we are not going to expand further on this issue. Instead, let us ask a provocative question: "Who writes genetic texts?" Since the time of Darwin, it has been acknowledged that "it is Evolution that writes genetic texts". Of course, Darwin and Mendel were not able to formulate the mechanisms of evolution as we understand them now, but they laid the basis for this, introducing the fundamental conceptions of evolution and genes. The modern molecular evolution theories claim that the original genetic texts, which existed billions years ago, have been evolutionary changed since then. The common way to formulate the idea of the molecular evolution is to say that *”Evolution is an adaptive process which transforms the genetic texts of organisms in such a way that they become more adapted to the environment they live in”.*

Evolution writes genetic texts according to certain laws, which are essentially different from those of the texts written by people. It can be said that, in the framework of these laws, not a single genetic text is written *de novo*; instead, one text is transformed to another by means of insertions, deletions or substitutions of letters as well as by transpositions of relatively large text fragments. *In*sertions and *del*etions are collectively referred to as *indels*. A fragment of one text may also be inserted into another text, the mechanism being referred to as *gene transfer*. Possibly, there exist still other mechanisms of genetic text transformations.

It should be emphasized that Evolution “rewrites” genetic texts very slowly: for example, in a human genome, approximately one nucleotide is changed (through a substitution, deletion, or insertion) in a year.

Below, we describe, the mechanisms of the genetic text evolution in more detail. The basic assumption is that every genetic text has at least one immediate predecessor (“at least one” means that one text can be generated from different predecessors by different series of elementary transformations). Currently, it is generally accepted that all modern genetic texts (the genomes of all living organisms on Earth) are descendants of a common ancestor. This idea was first formulated by Charles Darwin in *The Origin of Species* (1859) in the following form: *”...probably all of the organic beings which have ever lived on this earth have descended from some one primordial form, into which life was first breathed.”* However, at present, there is a controversy as to whether there was a single common ancestor or more. In any case, all genetic texts or, at least, large groups of texts must be interrelated.

It should be pointed out that we can fully observe only the genetic texts of currently existing organisms. As a rule, we do not suppose that one modern genetic text is an evolutionary offspring of another contemporary text. In view of the above considerations, it would be natural to assume that both texts have been generated from a single common ancestral text. Similar to the case of formal languages, we could say that two texts, $\mathfrak{T}'$ and $\mathfrak{T}''$, have both evolved from the same ancestral text $\mathfrak{T}$. Although text $\mathfrak{T}$ is not currently available, we can reconstruct it according to certain theoretical models. All these reconstruction models obey some common rules based on the formal assumption according to which all elementary text transformations constitute a basis with respect to mapping of a set of genetic texts onto itself.

However, since inverse mapping is polysemantic, the choice of a particular predecessor text from the range of all possible texts is strongly dependent on the model employed. The main objective of the reconstruction model is, actually, defining an inverse mapping of modern texts onto their predecessors. Thus, the reconstruction model narrows the range of possible predecessors to only one predecessor, converting an elementary text transformation to an unambiguous mapping of one text onto another. In other words, if we know that text $\mathfrak{T}_1$ has evolved from text $\mathfrak{T}_2$ through a series of elementary evolutionary transformations $t = (t_1, t_2, \dots t_n)$, the model enables us to reconstruct $\mathfrak{T}_2$ from $\mathfrak{T}_1$ unambiguously by means of $(t_n^{-1}, t_{n-1}^{-1}, \dots t_1^{-1})$. Unfortunately, we are not able either to reconstruct the actual succession of elementary transformations or to verify the employed model. Instead, we introduce an additive scoring function $(Sc(*))$ applicable to all elementary transformations considered in this book. Accordingly, the score of the series of elementery transformations t equals $Sc(t) = \sum\limits_{i=1}^{n} Sc(t_i)$. Obviously, text $\mathfrak{T}_1$ can be transformed to text $\mathfrak{T}_2$ by different series of elementary transformations. It is assumed that the most likely series is the one with the minimal penalty score Sc(t). Biological considerations play a major role both in the selection of the basic set of elementary transformations and in the assignment of the appropriate score to the elements of the set.

Historically, the proximity of two nucleotide sequences was first evaluated by calculating the number of local DNA transformations for a particular form of replacements. For example, the Jukes and Cantor model [124] assumes independent changes of all letters which occur with equal probability. According to this model, the proximity of two sequences is proportional to the number of letter replacements. Thus, the ACTG sequence is closer to the AATG sequence (one replacement) than to the ATT sequence (three replacements: a deletion and two insertion). The Kimura "two-parameter" model is almost as symmetric as that of Jukes and Cantor, yet it allows for a difference between *transition* and *transversion* rates. *Transversion* is a type of substitution where a purine is replaced with a pyrimidine(for example, GC with TA) or *vice versa. Transition* is a type of substitution in which a pyrimidine is replaced by another pyrimidine or a purine is replaced by another purine. In the current models, the distances between sequences are calculated on the basis of all three possible types of local transformations. Namely, one or both sequences undergo local transformations which "equalize" the sequences in a certain sense, the process being referred to as *alignment*. For example, the sequences ATCA and ATTA can be obtained from one another via two different series (i and ii) of local transformations (Fig. 4.2).

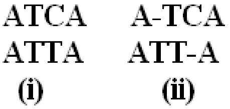


Fig. 4.2 Two possible ways of alignment of two sequences in the representation widely accepted in molecular biology. It is implied that the pairs of nucleotides which are located one above the other, correspond to the same nucleotides in the ancestor sequence. Thus, this representation emphasizes the evolutionary history of the sequences.

The first way (i) of alignment employs the transformation in which letter C of the upper sequence was replaced, in the course of evolution, by letter T (or *vice versa*, depending on the relative "age" of the two sequences). In any case, the distance between these sequences equals one elementary transformation.

The second way (ii) of alignment can be viewed as two different types of transformations. Namely, in one copy of the original ATA sequence, letter C was inserted after letter T, while in the other copy letter T was inserted after letter A. Thus, the "offspring" of a single common ancestor, ATA, comprises, in fact, two sequences - ATCA and ATTA, the distance between them being equal to two transformations (insertions).

The same result could be obtained in a different way - the sequence ATTA could lose one letter T (deletion) and acquire one letter C (insertion). In this case, the distance between the sequences also equals two elementary transformations. Since we are not aware of the actual ways of sequence evolution, it is usually assumed that the "actual" alignment is the one which is optimal with respect to a certain quality function. The simplest quality function is the number of elementary transformations required for sequence alignment: $\forall i, j : Sc(t_i) = Sc(t_j)$. Thus, in the above example, the "actual" way of alignment should be (i), since it corresponds to one elementary transformation. However, in certain models, the quality function may be equal to the sum of weighted elementary transformations. In this case, the actual way of alignment may be (ii) if the total weight of insertion and deletion (or two insertions) is less than that of one replacement.

The algorithm of evaluating the optimal alignment is based on the dynamic programming technique. By using this technique, the problem of optimal alignment is reduced to the search of the minimal-length way in a graph and can be readily solved.

On the other hand, non-local mechanisms of genetic sequence evolution have been known for quite a long time. They include transpositions of large blocks inside chromosomes or insertions of different mobile elements into genomes of higher organisms (eukaryotes).

Obviously, for such evolutionary mechanisms, alignment is not an adequate method of comparing genome sequences. Indeed, suppose that the sequence $\varPhi\varOmega\varTheta$ ($\varPhi$, $\varOmega$ and $\varTheta$ are sequences) has evolved into the sequence $\varPhi\varTheta\varOmega$. The alignment of these two sequences may result in the following scheme:

$$\begin{bmatrix} \varPhi\varOmega\varTheta- \\ \varPhi-\varTheta\varOmega \end{bmatrix}$$

i.e., in correct identification of merely two components, while the two sequences have much more in common. One can even formulate a certain "operational" criterion of alignment validity for nucleotide sequences: if the number of elementary transformations which are required for the alignment is relatively small, the alignment gives an informative result; otherwise, it is just a formal procedure.

4.4.1 Some Models of Sequence Evolution

The basic model for calculating the similarity between two texts is described in Section 4.5. The model is based on assuming the existence of a common predecessor for the two texts and a consequence of elementary operations that transform one text into the other. The latter consequence, actually, models a possible real evolutionary process if it meets certain optimization requirements, which are usually referred to as *minimal penalty score*. In the present section, we describe two simple models of the evolutionary process, which consider only nucleotide substitutions. These examples not only give the idea of the modeling technique, but also may be considered as basic since they have been sophisticated and generalized by means of a more adequate choice of the probabilities of nucleotide substitutions. In Appendix B, we describe the simplest version of the algorithm for finding the optimal sequence of transformations, which converts one text into another in the general case. This is, actually, the problem of finding the Levenshtein distance, considered above (Chapter 3, Section 3.1.1). This time, however, the interpretation of elementary transformations is different. In human language texts, substitutions and indels are considered to be errors; therefore, one of the two compared sequences is interpreted as the correct one, while the other is interpreted as being wrong. In the case of genetic texts, mutations which are fixed in the genome are no longer an errors, but rather urther evolution of the text. For this reason, both compared sequences are equitable.

4.4.1.1 The Jukes-Cantor Model

The Jukes-Cantor model [124] is the earliest and the simplest model of molecular evolution. It assumes that substitutions of all nucleotides occur independently and with equal probability. Moreover, a base can be substituted by each of the three other bases with equal probability. Table 1 shows the probability matrix for the mutation probability equal to 3adt, dt being a small time interval.

Let a certain position be occupied by nucleotide A. Evaluate the probability, $P_A(t)$, of nucleotide A occupying the same position at some time t.

Table 4.1 Probability matrix for nucleotide substitutions in the Jukes -Cantor model.

	A	T	C	G
A	1-3a	a	a	a
T	a	1-3a	a	a
C	a	a	1-3a	a
G	a	a	a	1-3a

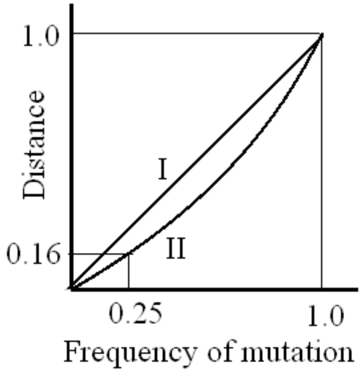


Fig. 4.3 Relationship between Hamming Distances (curve H) and Jukes-Cantor Distances (curve JC)

Direct calculation, which employs the substitution of the above probability matrix gives:

for $t = 0$, $P_A(0) = 1$;
for $t = 1$, $P_A(1) = 1 - 3a$;
for $t = 2$, $P_A(2) = (1 - 3a)P_A(1) + a[1 - P_A(1)]$.

The last expression results from two possible scenarios: (1) the nucleotide is not replaced; (2) some other nucleotide is substituted for A and, at t=2, the inverse mutation occurs. Thus, the following recurrent expression is true:

$$P_A(t + 1) = (1 - 3a)P_A(t) + a[1 - P_A(t)]$$

or

$$P_A(t + 1) - P_A(t) = -3aP_A(t) + a[1 - P_A(t)].$$

This equation could be solved directly, yet it appears preferable to use continuous time:

$$\frac{dP_A(t)}{dt} = -4aP_A(t) + a$$

The solution of the above differential equation, under condition $P_A(0) = 1)$, is

$$P_A(t) = 1/4 + (3/4)\exp(-4at).$$

For longer periods of time, the model gives the following probability of a nucleotide substitution (mutation) in the time interval t:

$$p = 1 - P_A(t) = \frac{3}{4}(1 - \exp(-4at)).$$

If the value of p is measured experimentally, it is worth inverting the above expression to obtain

$$at = -\frac{1}{4}\ln(1 - \frac{1}{4}p),$$

where p is the proportion of sites which are different in the two sequences. The value of at can be interpreted as the estimated distance, d, between the sequences.

Consider a simple example. Let two 100-nucleotide-long sequences differ in 25 nucleotides. The *Hamming distance* (the number of non-coinciding positions) between these two sequences equals, obviously, 25 or, in relative units, 0.25. However, the estimation in the framework of the Jukes-Cantor model gives the value of d=0.16. Fig. 4.3 shows the relationship between the Hamming distance and the Jukes-Cantor distance. It can be seen that the simplest consideration of the evolutionary mechanisms results in distances substantially different from the unsophisticated Hamming estimates.

4.4.1.2 The Kimura Two-Parameter Model

The Kimura model [138] is the next step towards a more realistic description of evolution. In this model, the probabilities of nucleotide substitutions are not symmetrical as in the Jukes-Cantor model. It is well known that substitutions of a purine for a purine or a pyrimidine for a pyrimidine (transitions) occur more frequently than substitutions of a purine for a pyrimidine or *vice versa* (transversions). Due to this fact, the probability diagram for nucleotide substitutions in the Kimura model has the form shown in Fig. 4.4. Table 4.2 shows the corresponding matrix of substitutions which occur in small periods of time. Obviously, it should be assumed that $1 - a - 2b > 0$.

The Kimura two-parameter distance between two sequences is

$$d = -\frac{1}{2}ln(1 - 2P - 2Q) - \frac{1}{4}ln(1 - 2Q),$$

where P and Q are proportions of sites that show transitional and transversional differences, respectively.

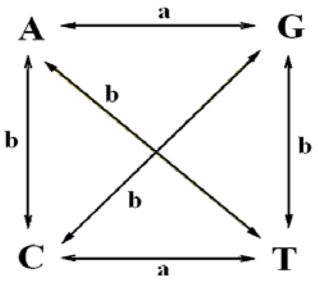


Fig. 4.4 Different rates of transitions (a) and transversions (b) in the Kimura model. A and G are purines, C and T are pyrimidines.

Table 4.2 Matrix of nucleotide substitutions in the Kimura model.

	A	T	C	G
A	1-a-2b	a	b	b
T	a	1-a-2b	b	b
C	b	b	1-a-2b	a
G	b	b	a	1-a-2b

4.5 Optimal Alignment of Two Sequences

The problem of alignment of two sequences in the Jukes-Cantor and the Kimura models can be reduced to considering only letter substitutions. From the standpoint of the algorithm, such substitutions can be readily localized, the specific difficulty of the problem being the evaluation of the substitution penalty scores. The latter are related to the probability of nucleotide mutations and the relevance of mutations to the biological function of the sequence. There exists a whole field of molecular biology which evaluates penalty scores on the basis of a huge body of sequence databases.

The models discussed above do not take into consideration indel-type mutations, because, from the "penalty point of view", indel mutations are relatively scarce (see above). The general model of the alignment of two sequences, which would account for mutations of all types and arbitrary penalty scores, could be effectively used only on the basis of rather sophisticated algorithmic techniques, which employ the method of dynamic programming. It has been already mentioned in this chapter that we are interested not in an arbitrary series of substitutions, which would convert one sequence into another, but only in those which correspond to minimal penalty scores. Provided the penalty scores for substitutions and indels are equal, the series we are looking for will be those of minimal length. In this section, we describe just such simplest model, which evaluates the minimal-length series of operations, converting one preset sequence into another preset sequence.

Table 4.3 Matrix M of letter correspondence for two sequences, $S_1 = \{AGGCCTGAG\}$ and $S_2 = \{AAGCTAG\}$.

	A	G	G	C	C	T	G	A	G
A	1							1	
A	1							1	
G		1	1				1		1
C				1	1				
T						1			
A	1							1	
G		1	1				1		

Table 4.4 Initialization of matrix $\tilde{M}$.

		A	G	G	C	C	T	G	A	G
	0	0	0	0	0	0	0	0	0	0
A	0									
A	0									
G	0									
C	0									
T	0									
A	0									
G	0									

Consider two sequences, $S_1 = \{AGGCCTGAG\}$ and $S_2 = \{AAGCTAG\}$, which are to be globally aligned. The lengths of the sequences are $|S_1| = 9$ and $|S_2| = 7$, respectively.

Consider now the similarity matrix M for these two sequences (see Table 4.3), of size $|S_1| \times |S_2|$. In this matrix, element m_{ij} equals 1 if the letter i in the sequence S_1 is the same as the letter j in the sequence S_2; otherwise, $m_{ij} = 0$. Next, let us define another matrix, $\tilde{M}$, of size $(|S_1| + 1) \times (|S_2| + 1)$, in the following way. To facilitate the calculations, we number the rows and the columns of the matrix starting from zero. First, the zero-number row and column are filled with zeros (see Table 4.4). Next step, matrix $\tilde{M}$ is successively filled in according to the following rule: each element $\tilde{m}_{ij} = MAX[\tilde{m}_{(i-1)(j-1)} + m_{ij}, \tilde{m}_{i(j-1)}, \tilde{m}_{(i-1)j}]$. Moreover, for each calculated value $\tilde{m}_{ij}$, the "direction" which has lead to this particular result is remembered. In other words, the program remembers which of the three values - $\tilde{m}_{(i-1)(j-1)} + m_{ij}, \tilde{m}_{i(j-1)}, \tilde{m}_{(i-1)j}$ - is the largest and thus, according to the algorithm rule, equals $\tilde{m}_{ij}$. For example, the value of $\tilde{m}_{11} = MAX[0+1, 0, 0] = 1$ is obtained as the sum $\tilde{m}_{(i-1)(j-1)} + m_{ij}$, which corresponds to the direction from the upper diagonal cell. The filled-in matrix is shown in Table 4.5.

Table 4.5 Filled-in matrix $\tilde{M}$, which is the basis for calculating the minimal sequence of elementary transformations.

		A	G	G	C	C	T	G	A	G
	0	0	0	0	0	0	0	0	0	0
A	0	1	1	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	2	2
G	0	1	2	2	2	2	2	2	2	2
C	0	1	2	2	3	3	3	3	3	3
T	0	1	2	2	3	3	4	4	4	4
A	0	1	2	2	3	3	4	4	5	5
G	0	1	2	3	3	3	4	4	5	6

Starting from the maximal penalty value, which equals 6 in this case, we build the inverse series of operations, passing from each cell to the one from which we arrived to the first cell when building the matrix $\tilde{M}$. The process can detect several forms of alignments, e.g.,

```
A G G C C T G A G
|   |   | |   | |
A A G _ C T _ A G
```

or

```
A G G C C T G A G
|   | |   |   | |
A A G C _ T _ A G.
```

However, the number of operations converting one sequence into the other is the same for all the alignments.

Finally, it should be noted that the minimum penalty alignment, actually, yields a metric on the sequence space. This result was obteined by Ulam [264] and also, for general gap functions, by Waterman *et al.* [279]. We do not cite the whole proof here, but just explain the arguments for the case considered by Ulam [264]. The crucial point is to show that the sequence distances obey the triangle inequality rule, i.e.,

$$d(x,y) \leq d(x,z) + d(y,z)$$

for three sequences, x, y, z, over the alphabet $\mathfrak{A}$. The minimal-distance alignment can be viewed as the minimum number of changes that convert one sequence into another. Assume that for three sequences, x, y, z, $d(x,z) + d(y,z)$ is less than $d(x,y)$. Consequently, the sequence z is a possible intermediate in the process of converting x into y. The distance between the latter two sequences can be $d(x,z) + d(y,z)$ in the worst case.

4.6 Attributes of DNA Sequences as Outcomes of Evolution Process

Using the information and communication terminology, it can be said that any genomic DNA text has evolved in a two-level process: text transmission and text change. Fragments of DNA texts (say, genes) may be transmitted from parents to offspring (at a short timescale) or from ancestral to descendant species (at a larger timescale). A much rarer event is *horizontal gene transfer* (or *lateral gene transfer*) - the event of transferring genetic material (a fragment of DNA text) from one organism to another which is not the offspring of the first one. On the other hand, a gene may be lost in the process of genetic information transmission from the ancestor to the descendant, the phenomenon being called a birth-death process [87].

Recent studies of DNA sequences have shown that different kinds of "textual" changes may occur in the course of genetic information transmission. Point changes, usually called *point mutations*, are *substitutions* (*transitions* and *transversions*), *insertions*, and *deletions*. Among point mutations, rates of substitutions are always higher than those of deletions or insertions. More global changes include *tandem duplication* of a text fragment, *inversion*, *translocation* and *transmission* between two sequences (*horizontal transfer*, *recombination*, and *conversion*). The process of the sequence change is superimposed on the process of transmission [87]. When two sequences are compared, it is impossible to tell whether a deletion has occurred in one sequence or an insertion has occurred in the other. Indels are not necessarily point mutations - the number of nucleotides in an indel may range from one to thousands. Indel lengths are characterized by a bimodal frequency distribution. One mode, which corresponds to short indels (up to 20-30 nucleotides), is caused by errors of DNA replication, while the other mode, corresponding to long indels, is mainly caused by recombination, transposition, or horizontal gene transfer.

Point mutations do not occur randomly throughout the genome. Some regions, called *hotspots*, are more likely to undergo changes; besides, transitions occur more frequently than transversions. For example, in animal nuclear DNA, transitions account for about 70% of all mutations (the expected rate of random transitions is 33%). In animal mitochondrial genomes, the ratio of transitions to transversions is about 20% . The rate of mutation is dependent on environmental conditions such as high radiation or extreme pollution. On the other hand, mutations are expected to occur with the same frequency regardless of whether they are beneficial or not to the organism. This aspect of the mutation justifies modeling it as a semantics-independent text change.

It might seem that different strains of the same bacterial species represent an example of genomes which must be reasonably similar to each other. However, bacteria were classified into species and strains on the basis of their phenotypical features, which does not necessarily imply the corresponding classification of the genomes. Since in bacterial systematics there is yet no agreement on the definition of a bacterial species [46], we employ the following "operational" definition: "A *bacterial species* may be regarded as a collection of strains that share many features in common and differ considerably from other strains."

Thus, the essential attributes of bacterial genomes should be those genomic "textual" properties that are shared by different strains of the species. One could suppose, for example, that such general genomic properties as the genome size, the nucleotide composition, and the number of genes correlate with the phenotypical strain classification. The data presented in Table 4.6 show that this supposition is wrong. One can easily notice significant differences in the above basic properties of the strains. Thus, these properties do not reflect the close evolutionary or structural relationships among bacteria. Furthermore, it was discovered that dinucleotide frequencies in natural DNA

Table 4.6 Basic properties of *Buchnera aphidicola* and *Escherichia coli* bacterial strains.

Genome	Length	Number of genes	GC content
	(Mb)	(proteins)	(%)
Buchnera aphidicola str. APS	0.64	564	26.3
Buchnera aphidicola str. Bp	0.615	504	25.3
Buchnera aphidicola str. Cc	0.42	357	20.1
Escherichia coli K12	4.6	4243	50.8
Escherichia coli O157:H7 EDL933	5.53	5324	50.4
Escherichia coli O157:H7 Sakai	5.5	5253	50.5
Escherichia coli UTI89	5.1	5044	50.6

deviate from those expected in random sequences. On the other hand, biochemical experiments [218] demonstrated that unrelated genetic sequences may have very similar dinucleotide frequencies. We can see, thus, that the above ” simple” approaches to genetic texts are not appropriate.

In Chapter 3, we described a few text-mining techniques that were developed for the characterization of human texts. In the following chapters, we present some applications of “linguistic tools” to evaluation of genetic texts. In the framework of such linguistic approach, a *word* is a basic linguistic unit that carries meaning and consists of one or more morphemes. A *morpheme* is the smallest meaningful unit of a language. The concept of morpheme differs from that of a word since, contrary to words, many morphemes cannot stand on their own. Such morphemes are called *bound*, while *free* morphemes can stand alone (have certain meaning). In synthetic languages, a single word stem (e.g., *love*) may have a number of derivatives (e.g., *loves*, *loving*, and *loved*), which are viewed as different forms of the same word. In such languages, words are considered to be constructed from a number of morphemes (in our example, the free morpheme *love* and the bound morphemes $-s$, $-ed$, and $-ing$). In most languages, words are easily identified due to word separators, which are most often spaces.

Having in mind to develop a linguistic approach to the analysis of genomic sequences, one can ask, *”What can be called a ”word” of a genetic text?”* In the DNA language, one type of a word/morpheme analog may be a *binding* (or *recognition*) site. A DNA binding site is a region of chemical bond formation with a specific protein. Such protein usually has a number of binding sites, which differ by indel- or substitution-type mutations in a few positions. For example, consider the following sequence of a recognition site - A[CT]N{A}, where only the base A is always found in the first position. [CT] stands for either C or T in the second position, N stands for any base, and {A} means any base except for A. It should be noted that the notation [CT] does not give any indication of the relative frequencies of C or T in this position.

In molecular biology and bioinformatics, the set of all binding sites of a particular protein is characterized by its *consensus sequence.* The consensus sequence is obtained by aligning all known “versions” of the recognition site and is defined as an idealized sequence that contains, in each position, the base that occurs in it most frequently. All the actual versions should not differ from the consensus sequence by more than a few substitutions.

By analogy with synthetic languages discussed above, different versions of a particular binding site may be viewed as derivatives of the same word (constructed of the same ”stem” and additional different morphemes). Therefore, in this example, a word of the DNA language is a consensus sequence for a particular protein binding site. Such a word, indeed, has a significant biological meaning.

Chapter 5
N-Gram Spectra of the DNA Text

It has been mentioned in Chapter 3 that text-mining techniques can be used to classify genomes. Out of all the methods considered in the previous chapter, the N-gram technique is one of the most appropriate for the genome text classification. In the field of linguistics, the N-gram concept has always been marginal and isolated. Similarly, in the case of genetic texts, a set of N-grams is in no way a set of functional elements. However, for the needs of formal text recognition, the N-gram technique proved to be exceptionally useful. On the other hand, the notion of "word" has not yet been successfully used in the genetic context. However, as it has been shown above, in this context, it is possible to give a definition of a "word" as having certain functional meaning. Nevertheless, the word as an element of the genetic text (similar to the case of hieroglyphic written language) is not as much flexible and universal as it can be in European languages.

5.1 Classification of Genomes on the Basis of Short-Word Spectra

5.1.1 Definitions

Let $\mathfrak{T}$ be a text over the alphabet $\mathfrak{A}$. In other words, $\mathfrak{T}$ is a sequence of letters from the given alphabet. For example, the DNA alphabet is $\mathfrak{A} =$ A,C, G, T. If we define an N-gram ξ of length N as a string of N characters over the given alphabet $\mathfrak{A}$, a text can be viewed as a stream of overlapping N-grams (see Fig. 5.1). A consecutive N-gram can be obtained from its predecessor by dropping its first character and adding the next letter T at its end. In the text $\mathfrak{T}$, there are $|\mathfrak{T}| - N + 1$ N-grams of length N. The length of a string S is denoted by $|S|$. N-gram techniques are based on the reduction of the whole text to one of the following vectors: the vector of N-gram observed frequencies, the vector of the observed/expected frequency ratios, or the vector of N-gram Usage Departures. For any N-gram ξ, let us denote the *observed*

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 61–85, 2010.
springerlink.com

```
...ATTCAAGCGCG...
 ATTC
  TTCA
   TCAA
```

Fig. 5.1 Overlapping 4-grams ATTC, TTCA, TCAA

frequency of the N-gram ξ in $\mathfrak{T}$ by $f(\xi, \mathfrak{T})$. The frequency of ξ is the number of ξ occurrences in $\mathfrak{T}$ divided by $(|\mathfrak{T}| - N + 1)$.

Given the observed frequencies of the N-grams of size (N-2) and of size (N-1), we can calculate the *expected* frequency of the N-grams of size N in $\mathfrak{T}$ as

$$E(a_1 \ldots a_N, \mathfrak{T}) = f(a_1 \ldots a_{N-1}, \mathfrak{T}) * f(a_2 \ldots a_N, \mathfrak{T}) / f(a_2 \ldots a_{N-1}, \mathfrak{T}). \quad (5.1)$$

In the case of N=2, the formula (5.1) is reduced to

$$E(a_1 a_2, \mathfrak{T}) = f(a_1, \mathfrak{T}) * f(a_2, \mathfrak{T}). \quad (5.2)$$

There exist 4^N different N-grams of length N. For example, for N=2 , there exist 16 N-grams: AA, AC, ..., TT. Given the observed frequencies f of bases x and y, the dinucleotide bias in $\mathfrak{T}$ is defined by the ratio

$$\rho_{xy}(\mathfrak{T}) = f(xy, \mathfrak{T}) / [f(x, \mathfrak{T}) * f(y, \mathfrak{T})] \quad (5.3)$$

and the *contrast value* is expressed as

$$q_{xy}(\mathfrak{T}) = f(xy, \mathfrak{T}) - f(x, \mathfrak{T}) * f(y, \mathfrak{T}). \quad (5.4)$$

In view of the peculiarity of the chromosome secondary structure, for each genetic sequence $\mathfrak{T}$, there is the complementary text $\mathfrak{T}''$ on the other string of the double-stranded DNA. In some applications, in order to accommodate the double-stranded DNA, it is necessary to introduce symmetrized *base* frequencies in the form

$$f^*_A = f^*_T = \frac{1}{2}(f('A', \mathfrak{T}) + f('T', \mathfrak{T})); f^*_C = f^*_G = \frac{1}{2}(f('C', \mathfrak{T}) + f('G', \mathfrak{T})) \quad (5.5)$$

and symmetrized *dinucleotide* frequencies in the form

$$f^*_{xy}(\mathfrak{T}) = f^*_{\bar{y}\bar{x}}(\mathfrak{T}) = \frac{1}{2}\left[f\left(xy, \mathfrak{T}\right) + f\left(\bar{y}\bar{x}, \mathfrak{T}\right)\right] \quad , \quad (5.6)$$

where $\bar{A} = T; \bar{T} = A; \bar{C} = G; \bar{G} = C$.

The symmetrizations of formulas (5.2) and (5.3) are, respectively,

$$\rho^*_{xy}(\Upsilon) = f^*_{xy}(\Upsilon)/[f^*_x(\Upsilon) * f^*_y(\Upsilon)] \tag{5.7}$$

and

$$q^*_{xy}(\Upsilon) = f^*_{xy}(\Upsilon) - [f^*_x(\Upsilon) * f^*_y(\Upsilon)]. \tag{5.8}$$

5.1.1.1 Linguistic Measure of Sequence Relatedness

One of the first attempts to reduce a long DNA sequence to a point in the space of n dimensions for further sequence comparison was made in ([203]). The authors suggested to use "*the linguistic similarity measure for fast and simple preliminary nucleotide sequence characterization and for estimation of relatedness to other sequences*". They knew that dinucleotide frequences alone could not adequately represent genomic sequences because they were aware of the biochemical experiments which demonstrated that unrelated genetic sequences might have very similar base composition or even dinucleotide frequencies. So, linguistic similarity was defined using the notion of *contrastwords* and *vocabularies*. The *contrastvalue* of each N-gram ξ in the sequence S is defined as the difference between its observed and expected frequencies: $q(\xi, S) = f(\xi, S) - E(\xi, S)$. Brendel *etal.* [31] suggested the term *contrast* $N - vocabulary$, V_N, for the set of the contrast values of all the N-grams of length N. Since the size of the DNA alphabet is equal to four, the size of the vocabulary $|V_N| = 4^N$. The sequence S of length $|S|$ is transformed into a point $||q_i||$ in the 4^N-dimentional space: $1 \leq i \leq 4^N$. Instead of the Euclidean distance between the points in such a space, the authors proposed to measure similarity by the following correlation coefficient formula:

$$C_N(S_1, S_2) = \frac{\sum_{j=1}^{4^N} q_j(S_1) q_j(S_2)}{\sqrt{\sum_{j=1}^{4^N} q_j(S_1)^2}\sqrt{\sum_{j=1}^{4^N} q_j(S_2)^2}}, \tag{5.9}$$

where $C_N(S_1, S_2)$ is the correlation coefficient between the sequences S_1 and S_2, N is the chosen length, 4^N is the number of all possible N-grams of size N, and $q_j(s)$ is the contrast value of the N-gram j in the sequence s. On the basis of empirical observations, Pietrokovski *et al.* [203] argued that the use of relatively small N $(2 \leq N \leq 5)$ could satisfy all practical needs of a researcher and proposed to use the integral value $C_{2-5}(S_1, S_2) = \frac{1}{4}[C_2(S_1, S_2) + C_3(S_1, S_2) + C_4(S_1, S_2) + C_5(S_1, S_2)]$ for the quantitative description of linguistic similarity between S_1 and S_2.

Applying the C_{2-5} measure to the data available at that time, the authors were able to demonstrate the power of their method by quite a few examples. They even declared that the vocabularies obtained from relatively short genetic sequences contained taxonomic "signatures" sufficient for their proper classification. Using these methods and employing $C_{2-3} = \frac{1}{2}(C_2 + C_3)$, Pietrokovski and Trifonov [202] identified imported sequences in the mitochondrial (*mt*) yeast genome in the following manner. Certain sequences from the *S. cerevisiae mt* genome were selected; various correlation coefficients $C_{2-3}(S_1, S_2)$ were calculated for one of the above-mentioned *mt* sequences, S_1,S_2 being either a nuclear or a *mt* long chromosome. Among the studied *mt* fragments, Pietrokovski and Trifonov identified several fragments with the contrast vocabularies significantly similar to the nuclear vocabulary.

5.1.1.2 Vector of Dinucleotide Relative Abundances

In the early 1990s, Sam Karlin proposed to reduce a long genomic sequence to a short vector by introducing dinucleotide relative abundance values, which he has examined since then [127], [130], [129], [132], [131], [132], [128].

The *dinucleotide relative abundance value* of each N-gram ξ in the sequence S is calculated as the ratio of its observed and expected frequencies: $q(\xi, S) = f(\xi, S)/E(\xi, S)$. As a measure of the *dinucleotide distance* between two sequences S and S', Karlin chose the $\delta - distance$ $\delta(S^{,}S')$, which is, actually, the Manhattan distance between the vectors $\rho^*(S)$ and $\rho^*(S')$):

$$\delta(S, S') = (1/16)\Sigma|\rho_i^*(S) - \rho_i^*(S')|. \tag{5.10}$$

Similar to the case of equation 5.7, $\rho_i^*(S)$ traverses all the dinucleotides in the range of $1 \le i \le 16$. Karlin *et al.* [131] noted that the transformation of a sequence to a vector of dinucleotide relative abundance values is essentially different from the transformation to the vector of frequencies. Thus the δ-distance 5.10 is in marked contrast to the non-normalized dinucleotide frequency based on the Manhattan distance:

$$d(S, S') = (1/16)\Sigma|f_i^*(S) - f_i^*(S')|. \tag{5.11}$$

Karlin and Mrazek [132] studied bacterial genomes in order to determine dinucleotide relative abundance values for DNA fragments. They found that the sequences obtained from the same prokaryote are clustered together, while the sequences obtained from distantly related bacteria are significantly separated from each other as judged by their dinucleotide relative abundance values. Thus it can be suggested that these values may be affected by some inherent genome factors such as replication and repair machinery functioning, overall genomic superhelicity (see Section 1.5 and species-specific mutation patterns. Karlin referred to the ensemble $\{\rho^*(xy)\}$ of all dinucleotides counted along a representatively long sequence of the genome $\mathfrak{G}$ as the *”dinucleotide relative abundance profile of the genome* $\mathfrak{G}$*”* or the *”genomic signature”*.

Having a method to measure dissimilarity between two genomes, one can investigate whether clustering based on δ-distances makes any biological sense. Some investigation was conducted by Sam Karlin and presented by him in numerous publications (see, *e.g.*, his review [127]). Several fragments were randomly chosen from a number of prokaryotic genomes and δ-distances between the nucleotide pairs were calculated. In particular, Karlin attempted to answer the question as to whether the Archaea genomes (see Chapter 2) constitute a coherent group under the conditions of δ-distance-based clustering. The answer to this question was negative [127]. Indeed, the genome of the Archaea *Halobacterium*[1] sp. was found to be very distant from those of Archaea *Sulfolobus*[2] and *M.Jannaschii*[3], while the *Streptomyces* sequences and some fragments of *M. tuberculosis* appeared to be the closest to the *Halobacterium.* These findings seem to be totally absurd from the phylogenetic point of view; however, they may be accounted for by certain biological effects. According to Karlin's global observation, thermophiles tend to be relatively closer to vertebrate eukaryotes than to Eubacteria, whereas *Halobacteria* sp. are very distant from vertebrates. It should be noted that, on the basis of compositional spectra, thermophiles and vertebrates (see Chapter 6) are also clustered together. Karlin [127] mentions many other discrepancies between δ-distance-based clustering and conventional phylogeny.

5.1.1.3 Survey of Genome Signatures in Prokaryotes

Many prokaryotic genomes have been completely sequenced since 1998. In order to assess the applicability of the genomic signature technique, Van Passel *etal.* [196] calculated a great number of prokaryotic genomic signatures using the Karlin's methodology (without any modifications). The authors compared the results of the genomic signature-based clustering with the 16S rRNA-based phylogeny (see Chapter 2) for 334 prokaryotic genome sequences. The δ-distances between all $334 * 333/2$ genomic pairs were calculated and the δ-distance values between chromosomes from the same genus and those of prokaryotes with multiple chromosomes were compared. *Genera* were defined as organisms with the same genus name, while *species* were defined as organisms with the same genus name and the same specific designation. Average intrageneric δ-distances for 40 different genera consisting of both Archaea and Bacteria indicated a large variation in the genomic signatures. Four genera showed pretty high dissimilarity scores, while five genera appeared to be extremely close.

The δ-distance values between chromosomes from the same species appeared to be very close; however, four exceptions were found. As we have mentioned above (Chapter 2), bacterial systematics has not yet reached a consensus on the definition of a bacterial species [46]. Anyway, conventional systematics considers some prokaryotes as strains of the same species, while

[1] The genus *Halobacterium* ("Salt" or "Ocean Bacterium") consists of several species of Archaea, which require an environment with a high salt concentration.

[2] *Sulfolobus* species grow in volcanic springs with the optimal growth temperatures of about $75 - 80°$C, which makes them thermophiles.

[3] *M. jannaschii* can grow in habitats with pressure up to more than 200 atm and temperatures ranging between 48 and 94°C, with the optimum growth temperature of 85°C.

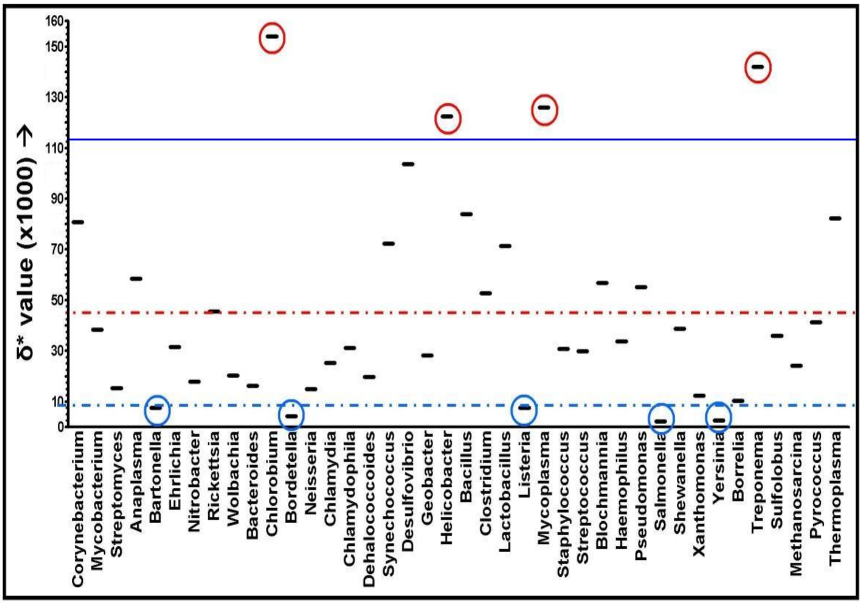


Fig. 5.2 Intrageneric δ-distances for 40 prokaryotic genera.

the δ-distance method would classify them as separate species. Van Passel *et al.* [196] concluded that δ-distances had strong phylogenetic relevance and could be used to support or oppose a given phylogeny and the resulting taxonomy.

5.1.1.4 Tetranucleotide Frequency Bias

The study of dinucleotide relative abundance values of prokaryotic chromosomes had a rather limited success. Pride *et al.* [207], [208] and Robins *et al.* [213] tested whether tetranucleotide-based clustering would be of greater phylogenetic relevance than the dinucleotide-based clustering discussed above. Pride *et al.* proposed to use *tetranucleotide usage departures from expectations* (TUD) - the vector of the ratios, $F(W)$, of the observed occurrence value, $O(W)$, of a tetranucleotide W to its expected occurrence value $E(W)$, i.e., $F(= O(W)/E(W)$. The ratios $F(W)$ are calculated for all 256 tetranucleotides. The expected value $E(W)$ is determined either from equation 5.12 or from equation 5.13:

$$E(W = w_1w_2w_3w_4) = f(w_1)f(w_2)f(w_3)|S|; F(W) = \frac{O(W)}{E(W)} \qquad (5.12)$$

$$E(W = w_1w_2w_3w_4) = \frac{O(w_1w_2w_3)O(w_2w_3w_4)}{O(w_2w_3)}; F(W) = \frac{O(W)}{E(W)}, \quad (5.13)$$

where w_i is the ith nucleotide of W; $f(\mathrm{A})$, $f(\mathrm{C})$, $f(\mathrm{G})$, and $f(\mathrm{T})$ are the nucleotide frequencies for the sequence S which is evaluated; $|S|$ is the length of the sequence and $O(X)$ is the observed occurrence of the N-gram X in the sequence S [31], [133], [225]. The calculation of the expected occurrences was discussed in detail in the previous chapter. It should be noted that Pride *et al.* [207] proved the validity of the Chargaff's second parity rule (see Section 1.2.1.1) for tetranucleotides in each of the analyzed genomes. The comparison of $O(W)$ for each tetranucleotide combination with the corresponding value for the reverse-complement of each combination, performed by means of linear regression analysis, yielded the correlation coefficient values of 0.99. The same effect was demonstrated in [141] for N-grams of length N=10. In numerous recent studies it has been shown that the Chargaff's second parity rule is valid in most cases, except for certain special chromosome regions.

To perform the cluster analysis based on TUD, Pride *et al.* [207] used the Manhattan distance. In this case, the distance between two organisms, 1 and 2, is calculated as

$$D_{1,2} = \frac{1}{256}||F_1(W) - F_2(W)||, \quad (5.14)$$

where $F_1(W)$ and $F_2(W)$ are vectors of departures for organisms 1 and 2, respectively, obtained in the way similar to that of Cardon and Karlin [133] (see equations 5.12 or 5.13). Pride *et al* [207] used the distances between tetranucleotide frequencies calculated on the basis of the zero-order Markov model in order to construct a phylogenetic tree for 27 bacterial genomes. The authors concluded that the TUD patterns carry a phylogenetic signal. Teeling *et al.* [244], [245] found that the above-mentioned results could be improved by using whole-genome sequences as a distance measure. The expected tetranucleotide frequencies were calculated on the basis of the maximal-order Markov model (5.11). Next, the divergence between the observed and the expected tetranucleotide frequencies was estimated using the z-scores (see equations 5.15, 5.16) and the approximation published by Schbath *et al.* [225], [224]:

$$Z(W = w_1w_2w_3w_4) = \frac{O(w_1w_2w_3w_4) - E(w_1w_2w_3w_4)}{\sqrt{varO(w_1w_2w_3w_4)}}, \quad (5.15)$$

whereby the variance $var(O(W))$ can be approximated as follows:

$$varO(W) = E(W)\frac{|O(w_2w_3) - O(w_1w_2w_3)||O(w_2w_3) - O(w_2w_3w_4)|}{O(w_2w_3)^2}. \quad (5.16)$$

The question as to whether two genomic fragments exhibit similar patterns of over- and underrepresented tetranucleotides can be addressed by calculating the Pearson correlation coefficient for their z-scores. Similar patterns correlate with high correlation coefficients, whereas diverging patterns have low correlation coefficients.

By means of clustering based on tetranucleotide frequency vectors and on the above-mentioned distances between the vectors, the congruence between the genome trees obtained by the conventional and by the N-gram-based methods can be detected. Despite the presence of an evident phylogenetic signal in the tetranucleotide frequencies, the methods proposed by Pride *et al.* [207] and by Teeling *et al.* [244], [245] failed to reconstruct large phylogenetic trees in a manner that would conform with the standard 16S rRNA-based topology. It was found that closely related species were correctly grouped in most cases, whereas more distant species often formed combinations opposing the "phylogenetic logics". The authors concluded that distant relationships cannot be evaluated on the basis of TUD patterns; partitions based on such vectors only partially carry phylogenetic signals.

5.2 Fuzzy *N*-Grams and Compositional Spectra of Sequences

In this chapter, the reader is introduced to the method which allows for both local and global evolutionary mechanisms when being applied to the comparison of nucleotide sequences [140], [141], [139], [197]. Consider a simple example. The sets of all possible 10-letter N-grams for two sequences, ACGTTGACTTGG and AtGTTGACgTGG (small letters stand for replacements), are

$$N_1 = \{ACGTTGACTT, CGTTGACTTG, GTTGACTTGG\}$$

and

$$N_2 = \{AtGTTGACgT, tGTTGACgTG, GTTGACgTGG\},$$

respectively. Although, formally speaking, these N-grams are not identical, they may be considered evolutionary close since the difference between them is of the order of one or two transformations (replacements, in this particular case). In other words, the N-grams which belong to the sets N_1 and N_2 have the same frequency of occurrences in both sequences (with regard for alignment). Thus, both sequences appear to be close, which is quite natural for the situation under consideration.

Given a certain sequence S and a certain N-gram w, the N-gram can be located within the sequence using the idea of alignment (Fig. 5.3).

In fact, the alignment is performed between the underlined word and w. Transformations of the replacement type occur in the indicated boxes. The example (i) shown in Fig. 5.3 bears, in a sense, negative connotation.

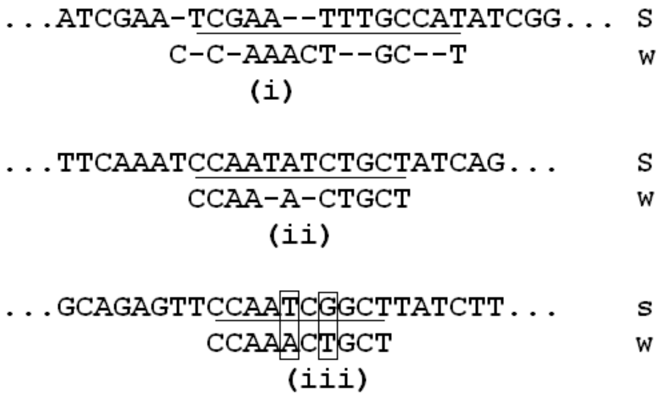

...ATCGAA-TCGAA--TTTGCCATATCGG... S
C-C-AAACT--GC--T w
(i)
...TTCAAATCCAATATCTGCTATCAG... S
CCAA-A-CTGCT w
(ii)
...GCAGAGTTCCAATCGGCTTATCTT... s
CCAAACTTGCT w
(iii)

Fig. 5.3 Different ways of locating the N-gram, w, within the sequence S using the alignment strategies.

Although w has really been located in the sequence S, nine insertions and deletions (indels) were used in the process, while the length of w is 10. This example rather shows that any N-gram can be inserted into the sequence S starting from any position under the condition that any required number of elementary transformations is permissible. Obviously, such unlimited perception of "inclusion into sequence" makes this concept meaningless. In the next example, (ii), only two indels are required to include w into S. Although we have not defined the weight function of these transformations and, as a result, are unable to score the alignment formally, it appears, from the standpoint of meaningfulness, that we have found the proper position in the sequence. In the last example,(iii), only two replacements are used for the alignment. In what follows, we restrict ourselves to using the transformations of the replacement type; however, the main definitions are formulated in the general form.

Definition 1. *We assert that the N-gram y has an imperfect occurrence in the sequence S if there exists such a substring x of S that the distance between x and y is less than a predefined threshold (in the given metric).*

This definition goes far beyond our current needs and allows us to substantially generalize the concept of the N-gram occurrence in a sequence and, consequently, the concept of the compositional spectrum [140]. However, still remaining in the framework of the elementary transformation model, we will constrict this definition to the following one:

Definition 2. *The N-gram y has an imperfect occurrence in the sequence S if there exists such a substring x of S that the smallest weighted sum of all the replacements, insertions, and deletions between y and x is less than or equal to a given threshold value r.*

This sequence metric, which is well known in sequence-alignment applications [278], is discused in detail in Chapter 4.

If the alignment is performed using only replacements and the latter have equal weights, the previous metric is reduced to the Hamming distance and Definition 2 is converted into:

Definition 3. *N-gram y has an imperfect occurrence in S if there exists such a substring x of S that the Hamming distance between y and x is less than or equal to a given threshold value r.*

This approximate matching can be denoted as "r-mismatch".

Let us consider a set W of n different N-grams w_i of length N. The number of imperfect occurrences of w_i in the target sequence S is $m_i = occ(w_i|S)$. Now let $M = \Sigma m_i$. The frequency distribution $F(W,S) : \{f_i = m_i/M\}$ will be referred to as the *compositional spectrum*[140] of the sequence S relative to the set W.

5.2.0.5 Calculation of a Compositional Spectrum

Possible algorithms of the calculation of CS will be discussed below starting with the simplest case which corresponds to Definition 3. In this case, the calculation of imperfect N-gram occurrences in the sequence S does not require any sophisticated algorithmical procedure. Indeed, according to the definition, we look over all possible substrings x of size N. The number of N-gram occurrences in S is equal to the number of such substrings x that the Hamming distances from x to the particular N-gram do not exceed the preset level r. Obviously, the number of operations in such an algorithm is of the order of $N|S|$ since the length of the sequence S is $|S|$ and, for each N-gram, N comparisons are executed.

Since the spectrum is calculated over a certain set W of N-grams, the total number of operations for calculating the spectrum is of the order of $N|S||W|$, where $|W|$ is the number of elements in the set W. Thus, the calculations which employ such a simple algorithm take a considerable amount of time.

The problem of looking for algorithms which allow to calculate compositional spectra much faster, in the case of a perfect N-gram occurrence in the sequence, *i.e.*, at the zero mismatch level (r=0), has been addressed in a large body of research. Below we will mention just a few of the existing algorithms, which are listed in Table 5.1.

The Brute Force algorithm (line 1) was, in fact, described by us at the beginning of this Section.

The Knuth-Morris-Pratt algorithm, in its basic form, which is specified in Table 5.1 (line 2), is very similar to the Brute Force algorithm.

The Karp-Rabin algorithm (line 3) employs two concepts - hashing and windowing. Hashing is implemented by generating a "key" for the pattern being searched; the subsequent windowing stage is performed by comparing

Table 5.1 Some algorithms of searching for perfect occurrences of a substring in a string.

Algorithm	Pre-processing	Average time of search	The worst time of search	Memory volume
Brute Force algorithm		2*$\|S\|$	O($\|S\| * N$)	
Knuth-Morris-Pratt algorithm	$O(L)$	$O(\|S\| + N)$	$O(\|S\| + N)$	$O(L)$
Karp-Rabin algorithm	$O(L)$	$O(\|S\| + N)$	$O(\|S\| * N)$	
Boyer-Moore algorithm	$O(N + 4)$	$O(\|S\| + N)$	$O(\|S\| * N)$	$O(N + 4)$

this "key" with the values generated by moving a "pattern-sized" window along the target text.

The Boyer-Moore algorithm scans the pattern characters from right to left. When a mismatch is found, the window is shifted to the right using two precomputed functions, which are referred to as the *"good-suffix shift"* and the *"bad-character shift"*.

A full description of the above algorithms can be found in [5], [4].

The well-known names which appear in Table 5.1 and a large number of algorithms imply that the problem of constructing the time-optimal algorithm for searching perfect occurrences of a substring in a string is quite difficult. Finding the optimal algorithm to determine whether a substring occurs in the string imperfectly is a still more complicated problem. There exist quite a few algorithms which approach this problem in different ways.

For example, one can use the method of Landau and Vishkin [161] or its modification [85] to find, during the time $O(|S|r)$, all locations along the sequence S where the N-gram coincides with the text with the allowed mismatch r. These algoritms use the suffics tree for the problem of string matching. The advantage of the Abrahamson algorithm [1] is that its run time is independent of r. The algorithm developed in [10] is always faster than all the above-mentioned algorithms. According to this method, at the first, marking, stage, the text is marked in such a way that the position number of a letter of the text is connected with the position number of the same letter in the N-gram. At the second, verifivation, stage, the marks are used to determine all the locations where the N-gram coincides with the text with the mismatch not larger than r.

The Brute Force algorithm can also be improved (through the use of preprocessing) to be applied to the same problem of searching for imperfect occurrences of substrings. Preprocessing can consist in lexicographical ordering (similar to word ordering in a dictionary) of the N-grams which belong to the vocabulary W. The comparison of an N-gram with each substring of length

L of the sequence S employs about $N * ln(|W|)$ operations, which results in approximately $N|S|ln(|W|)$ operations required for the calculation of the spectrum. Similarly, preprocessing can be performed through lexicographical ordering of all the sub-strings of length N of the sequence S. According to this method, each substring gets a certain index, which is equal to the number of the substring copies in the sequence S. In this case, the comparison of each N-gram of the vocabulary W will require $N * ln(|S|)$ operations and, consequently, the calculation of the whole spectrum will require $O(N * ln(|S|)|W|)$ operations. The choice of a particular preprocessing depends on the data parameters.

Another algorithm with preprocessing can be effective in the case of relatively small N-gram lengths (about 10-15). First, for each N-gram w_i of the preset vocabulary W, we obtain the set, V_i, of all possible N-grams of length N, which differ from w_i by no less than the preset mismatch level r. This preprocessing, which is directly connected with the vocabulary, is the first step of the algorithm. The second step is the evaluation of the perfect occurrence frequencies for the preset sequence and for all possible N-grams of length N. This step can be implemented in different ways.

Version 1. At the preprocessing stage, we reduce the problem of imperfect occurrence of relatively small number of N-grams to that of perfect occurrence of a large number of N-grams. Therefore, it is possible to apply the algorithms of searching for N-gram perfect occurrences, *e.g.*, those from Table 5.1.

Version 2. The original 4-letter alphabet is encoded using numerical symbols 0,1,2,3. The sequence S is scanned with the N-length window, the unit shift being one symbol. Each time, the contents of the window is used as the address of the N-dimensional array and the corresponding cell of the array increases by one. As a result, for each N-gram of length N, the number of its perfect occurrences in the sequence is obtained. For example, let us suppose, for the sake of simplicity, that N=3 and the sequence S has the form S=1002311... in the digital code. Let this sequence be written in the array $S[i]$ so that $S[1] = 1$, $S[2] = 0$, $S[3] = 0$, $S[4] = 2$, $S[5] = 3, \ldots.$ All possible words of length 3 are associated with the three-dimentional array $v[0 : 3, 0 : 3, 0 : 3]$ in such a way that the array element $v[i_1, i_2, i_3]$ corresponds to the word i_1, i_2, i_3. For each word, the number of perfect occurrences is calculated using the following simple program cycle:

```
For i=1 to $|S| - 3$
    $v[S[i + 0], S[i + 1], S[i + 2]] = v[S[i + 0], S[i + 1], S[i + 2]] + 1$
Next i
```

At the last stage, the frequencies should be summarized over each of the sets V_i, which yields N-gram imperfect occurrence frequencies for the preset vocabulary and mismatch level.

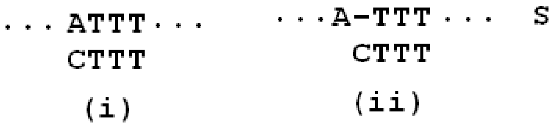


Fig. 5.4 Two different possible ways of the occurrence of the N-gram CTTT in the sequence S. (i) and (ii) dispositions differ by a one-position shift.

Let us consider now a more complicated problem of calculating a CS defined on the basis of the more general Definition 2. According to this definition, a particular N-gram occurs in a sequence if, in this sequence, there exists such a substring x that the alignment-based distance between x and the N-gram is not more than the preset level r. It should be noted that the distance of any other substring which contains the substring x does not exceed r, either. This fact does not contradict Definition 2; however, in order to calculate the total number of N-gram occurrences in the sequence S, we need, obviously, an additional definition. The following example demonstrates that, directly applying Definition 2 to calculating the number of N-gram occurrences, one may overestimate it. Fig. 5.4 shows two different possible ways of the occurrence of the N-gram CTTT in the sequence S.

The alignment of the substring ATTT (i) can be performed by substituting C for A (or *viceversa*). The alignment of the substring TTT (ii) requires a proper insertion. In both cases, only one elementary transformation is required and thus, according to Definition 2, each case signifies that the N-gram CTTT occurs in the sequence. Taking these two cases into account independently, one arrives at conclusion that the N-gram CTTT occurs in the sequence twice. However, it is obvious that it can actually occur only once in the considered fragment of S. Therefore, the correct way of reasoning should be the following. Let us consider the set $\mathfrak{X}(N\text{-gram}, S, r)$ of all the substrings which can be aligned with the N-gram using no more than r elementary transformations. On the set $\mathfrak{X}(N\text{-gram}, S, r)$ we define partial order with respect to inclusion in such a way that $a > b$ if $a \supset b$, $a, b \in \mathfrak{X}(N\text{-gram}, S, r)$. The set of minimal elements in this order is designated as $\mathfrak{X}_{min}(N\text{-gram}, S, r)$. Now we can represent the value Q of the N-gram occurrences in the sequence S as the number of the elements of this set: $Q = |\mathfrak{X}_{min}(\mathrm{N}{-}gram,\mathrm{S,r})|$. In the above example, the substring TTT represents, obviously, the minimal element and thus the N- gram CTTT occurs only once in the sequence fragment under consideration.

Now we can outline the Brute Force algorithm for calculating a CS on the basis of Definition 2. First, it should be noted that it is impossible to predefine the length of a substring of the sequence S, which the N-gram of length N should be compared to. If, in the process of alignment, all r transformations appear to be insertions in the substring, the initial substring length may be $N - r$. If all the insertions occur in the N-gram, the substring length may be $N + r$. Of course, all the intermediate lengths should also be considered. Next, according to the above definition, we look over all possible sub-strings of S in the length range from $N - r$ to $N + r$ and perform the procedure of alignment of each substring with the N-gram. If the alignment requires no more than r elementary transformations, the substring is included in the set $\mathfrak{X}(N\text{-gram}, S, r)$. Obviously, the number of operations in such an algorithm is of the order of $(1 + r)N|S|$. Next, we find the minimal elements in the set $\mathfrak{X}(N\text{-gram}, S, r)$ with respect to the established partial order. By definition, the number of these elements is equal to the number of N-gram occurrences in S. The total number of operations required for the calculation of the spectrum is, obviously, of the order of $O((1 + r)N|S||W|)$.

More complicated algorithms make it possible to solve the problem with better estimation of the number of required operations. For example, using the result obtained in [162], [160], we arrive at the estimate $O(r|S||W|)$ for the number of operations in the case of the mismatch r, the allowed operations being both substitusions and indels. Ukkonen [262] and Galil and Park [86] developed improved algorithms for the cases of a given bound on the number of allowed extended substitusions and indels. In a recent paper, Cole and Hariharan [48] suggested an algorithm with the estimate $O(|S| + (|S|r^4)|W|/L)$ for the case of r allowed substitusions and indels. Note that for small r the algorithm is linear.

5.2.1 CS Visualization and Some Aspects of Compositional Spectra Qualitative Analysis

The CS method is successfully applied to the problem of sequence comparison. This comparison will be quantitatively considered by us in the next Section 6, the consideration being not just formal, but rather performed with regard to qualitative features that are shared by CS of different sequences. In this connection, the CS visualization approach is of major importance, as it allows to qualitatively estimate the correspondence between the spectra of different sequences and observe the transformation of visual differences to formal numerical relations. At this point, it is worthwhile reminding that our field of science is essentially biological and, as such, may be investigated by a method of observation as well as by calculations [140], [141], [139].

Let S_1, S_2, ..., S_k be the genomic sequences that we are going to compare using the given set W. The CS, which is, actually, a frequency distribution on the set W, may be represented as a distribution plot, where X-axis corresponds to the running index j of N-grams w_j, while Y-axis corresponds to the frequencies f_{ij} of w_i. It is clear that the preset order of N-grams w_j in W predetermines the shape of the CS of S_i with respect to W. Of course, the spectra of different sequences can be visually compared only if the N-grams from the set W have the same order along the axis. For better visualization of multiple spectra, one can choose to order the N-grams $w_i \in W$ non-randomly, namely, the N-grams may be ordered by descending frequencies f_{ij} using any

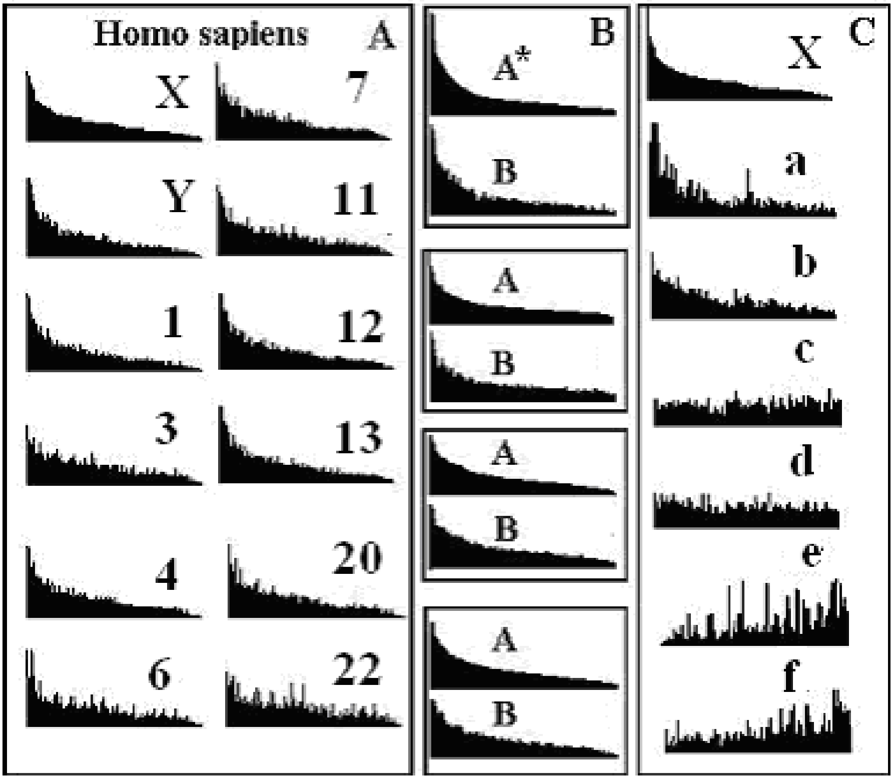


Fig. 5.5 Examples of compositional spectra obtained for various species on the basis of a particular N-gram set, W. The length of each N-gram, N, is 10; the number of N-grams, n, is 200; mismatch, r, is 2. **A** - spectra of long contigs from human chromosomes (C+G content is in the range 0.36 - 0.50), ordered with respect to the human X-chromosome (the specra numbers correspond to those of human chromosomes). **B** - spectra of different contig pairs from four genomes, ordered in each pair with respect to the first contig (A) (from top to bottom: *Caenorhabditis elegans*, *Drosophila melanogaster*, *Saccharomyces cerevisiae*, *Streptococcus pyogenes*). **C** - spectra of 7 contigs (**a** - *Caenorhabditiselegans*, **b** - *Saccharomyces cerevisiae*, **c** - *Escherichia coli*, **d** - *Porphyromonas gingivalis*, **e** - *Mycobacterium tuberculosis*, **f** - *Halobacterium sp.*, plasmida), ordered with respect to the human X-chromosome with different levels of similarity to the X-chromosome.

S, say $S = S_l$. In other words, we can denote by $Ord(W, S)$ such an arrangement that the N-gram w_i is followed by the N-gram w_j $(i > j)$ if and only if $m_i \geq m_j$ $(m_i = occw_i|S_0))$. In the case $m_i = m_j$, any reciprocal order of w_i and w_j is permitted. This order of N-grams w_i in W is non-random and can be derived from the target sequence composition. Such ordering facilitates the comparison of a given set of spectra with any particular spectrum (see the CS of a few species in Fig. 5.5).

One can see that all the considered contigs of human chromosomes display approximately the same spectrum (Fig. 5.5A) in terms of relative N-gram occurrence frequencies. However, the shapes of these spectra are quite different. For example, the frequency profiles for the chromosomes 1 and 3 decline in a similar way, which is quite different from the profiles for the chromosomes 6 and 11. The same observation is true for the spectra of different contig pairs from the four genomes shown in Fig. 5.5B, where the first contig (A) in each pair is ordered. This observation prompts the question as to the qualitative parameters which should be the basis for the quantitative comparison of the spectra. Keeping in mind the possible applications to intergenomic comparison, we select such quantitative measure which emphasizes the similarity of human contigs as parts of the same genome. Our visual estimation accentuates similarities on the basis of word order, while the estimation based on the spectra shapes should rather not be used. An example of a measure which depends only on the reciprocal order and is independent of the amplitude is the Spearman coefficient. At the same time, the common Euclidian distance measure is highly sensitive to the absolute values of the amplitude. The distance choice which would be appropriate for the genome classification problem will be discussed in detail in the next Section.

5.2.2 *N-Grams and Zipf's Law*

Passing on to the comparison of different genomes, let us consider the spectra of seven genomes (Fig. 5.5C), ordered with respect to the human X-chromosome. It can be seen that the spectra of different genomes are rather variable - they may be similar, neutral, or even opposite to each other. This fact demonstrates high resolution of the spectra over the whole set of genomes at the level of the N-gram order.

The shapes of the spectra have certain peculiarities. If we chose to draw the analogy with natural languages (as we often do throughout the book), we could suggest that the spectra shapes follow the well-known Zipf's law, which states that *"the probability of a word in a text, multiplied by the rank of its frequency, is a constant"*. Since it is the occurrence frequencies that are ordered (ranked), there is only one value of the occurrence frequency which is associated with a group of words having the same occurrence frequency. Zipf's law has the form $if_i = C$, where f_i is the occurrence frequency of the word in the text; i is the rank of the frequency; C is an empirical constant, which is chosen on the basis of the requirement that Zipf's relationships for all the frequencies be as close to equlities as possible. Suppose, for example, that the word *the* is the most frequent word, occurring in the text 17,186 times. Then, we would expect the second-most frequent word *a* to occur 8593 times (17,186/2). Similarly, we would expect the third-most frequent word *of* to occur 5729 times (17,186/3), the fourth-ranked word *to* to occur 4297 times

(17,186/4), and so on. As a result, Zipf's law is represented graphically by a hyperbola with a very long tail, accounted for by low-frequency words.

Actually, Zipf's law is a general synergetic law and is, therefore, applicable not only to genomic texts and natural languages, but also to a lot of other fields. For example, Zipf's distribution describes such varied events as scientific article citing, family name occurrences, the power of earthquakes, and the area of forest fires. It should be emphasized that, from a purely formal point of view, if a rank distribution is described by an exponential (Zipf-like) function, the distribution of the corresponding random value also follows an exponential law. A great number of empirical distributions found in nature, economics, sociology, and other fields of science, obey the exponential law. Noteworthy, that this empirical fact can sometimes be derived from fundamental physical conceptions as well.

Zipf's law was first discovered in a research in the field of linguistics [293], but, in that case, the words were considered as merely elements of a statistical ensemble. The same law would be true for the word distribution in a text typed by a monkey, which presses the keys on the keyboard in a random way [171]. The really surprising thing is that a text in a natural language, which is primarily aimed at conveying a particular sense, also demonstrates synergetic effects, if, of course, this text is not poetry. In conclusion, an important pertinent fact should be noted: Zipf's law also holds for DNA nucleotides, their pairs, triplets, and higher-order groups [178], [183].

Let us consider now the standard version of Zipf's law in more detail. If all the words of the text have different frequencies, the constant C can be readily obtained from the following obvious equation:

$$\sum f_i = 1 = C \sum \frac{1}{i}. \tag{5.17}$$

Indeed, for each i, according to Zipf's law, $if_i = C$, so that $f_i = \frac{C}{i}$. The subsequent summarizing gives (5.17) since the total sum of all the word frequencies equals 1.

It is well-known, however, that the constant C is different for each language, which is due to two reasons. 1) There always exist groups of words which have the same occurrence frequency, so that the sum of the frequencies (of course, without regard for multiplicity) is less than 1 and cannot be found *a priori*. 2) The sum of inverse rank values in the right part of (5.17) depends on the number of various words in the text, the number being different for different texts.

Now it will be shown that Zipf's law may, indeed, be applied to the description of the shape of compositional spectra.

The data presented in Table 5.2 suggest that Zipf's law may adequately describe the CS shapes since the calculated values of XY are approximately constant (300 and 200) for both chromosomes.

Table 5.2 Some N-gram frequencies (Y-coordinate) and positions (X-coordinate) obtained from the X-chromosome spectrum (Fig. 5.5A) and from the A^*-spectrum (Fig. 5.5B).

Fig. 5.5A, X			Fig. 5.5B, A*		
X-coordinate	Y-coordinate	XY	X-coordinate	Y-coordinate	XY
12	25	300	8	25	200
19	15	285	15	15	225
31	10	310	31	7	217
46	8	368	52	6	312
71	4	284	80	3	240

5.2.3 Distances between Compositional Spectra

In the linguistic text theory, there exist a lot of various distance types, or, in a more general sense, text dissimilarity measures. However, their efficiency is not of universal nature, being connected with the peculiarities of a text content. Naturally, in the case of genomic texts, other preferred distances may be chosen. In this paragraph, we test certain possible types of distances between CS with due regard for specific problems of genome classification.

By definition, the *compositional spectrum* of the sequence S relative to the set W is a vector of frequencies $F(W, S)$. There are various methods of measuring the dissimilarity between two distribution-vectors, $t = (t_1, t_2, ..., t_n)$ and $u = (u_1, u_2, ..., u_n)$. In what follows, several distances considered in Appendix A, Section A.1.2 will be compared: the Euclidian distance, the Manhattan distance, the Max-distance, the KS-like distance, correlation dissimilarity, the Spearman dissimilarity, and the Kendall dissimilarity. Obviously, the effectiveness of using distances for the analysis of genomic sequences cannot be assessed theoretically and should be determined “empirically”. The following example shows how the discriminating ability of various distances can be assesed [274].

Example. Consider 38 particular species, which represent three kingdoms of life (*Archae*, *Bacteria*, and *Eucaria*). Each species is represented not by the full genome, but by two different genome sequences, 300-400 kb long. Thus, the initial set (database) includes 76 sequences. Such structure of the database (more than one sequence from each genome) allows to control the grouping quality of the data.

Over the set of all the database sequences, each of the above-mentioned functions generates a set of pair-wise distances. It is convenient to describe each set by the corresponding histogram.

According to the obtained histograms, all the distance measures can be divided into two groups. The first group includes the Euclidian, Manhattan, Max-, and KS-like distances. The second group contains the dissimilarities based on the Pearson, Spearman, and Kendall correlations. The Euclidian

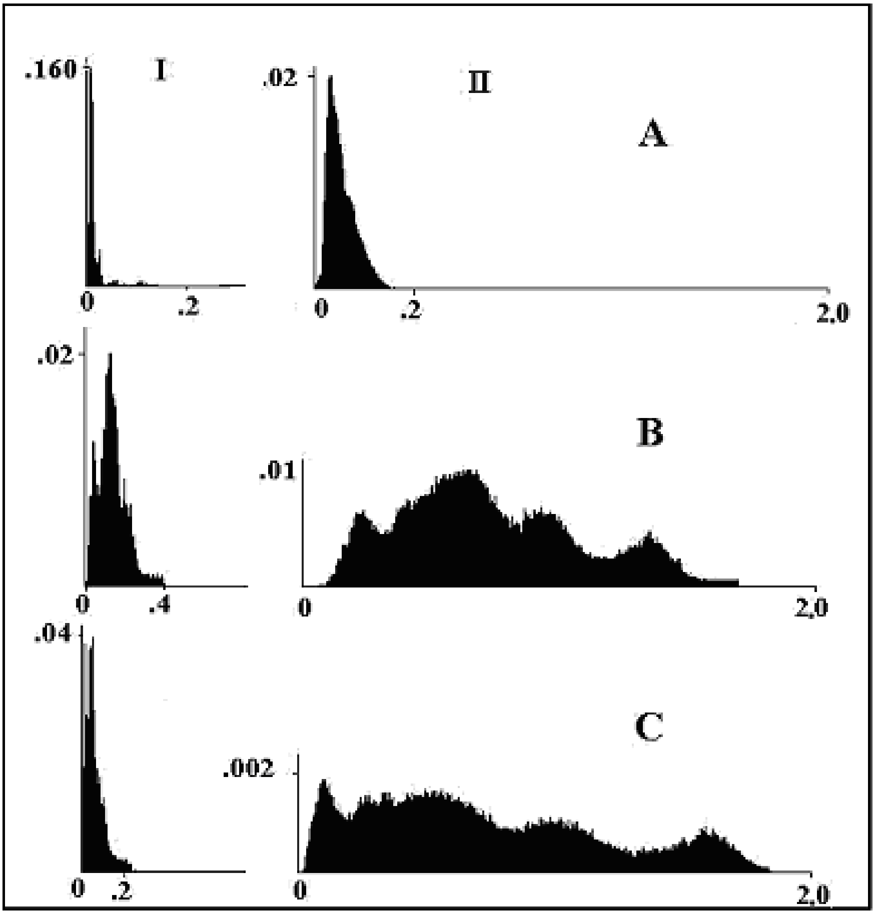


Fig. 5.6 Distribution of CS-distances based on three different distance measures. **A** - the Euclidian distance; **B** - the Kendall rank correlation τ; **C** - the Spearman rank correlation ρ. The first column - the distributions of intragenomic distances, the second column - the histograms of intergenomic distances. The N-gram length N=10; the number of N-grams n=200; mismatch r=2.

distance (Fig. 5.6A) provides a typical example of a distance histogram for the first group. A histogram of such type suggests the existence of a single cluster. ensemble

An example of a histogram for the second group is shown in Fig. 5.6B,C, where the distances were calculated using the Spearman and Kendall rank correlation coefficients.

Three major local minima can be observed on the histograms, but the disparities between the masses of the maxima are not so considerable as those for the first group of distances. This result suggests the existence of three or

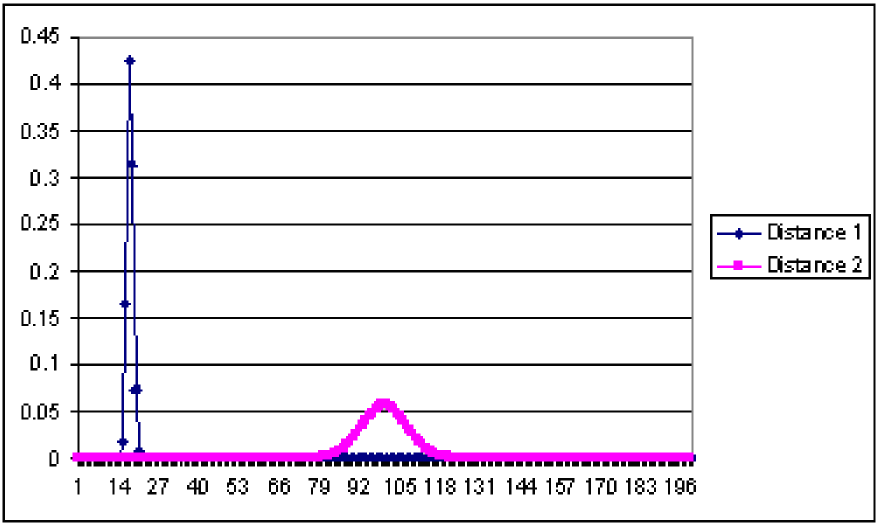


Fig. 5.7 Distribution of the Euclidian distances and of the distances based on Spearman correlation coefficients over the set of 1,000 randomly generated artificial CS. 1 - Euclidian distances; 2 - distances based on Spearman correlation coefficients. The number of N-grams n=200.

more clusters. It can be concluded, thus, that the second group of measures is preferable for effective clustering. This group includes also a measure based on the N-gram vector correlation, which is often employed to build distance matrices.

Thus, using a CS set calculated for natural genomes, it can be shown that the pairwise distance distribution substantially depends on the choice of the distance function. However, this distribution also depends on the vector distribution in the corresponding space, which, in turn, reflects the genomic structure peculiarities. Indeed, consider a random vector, which is called random since each of its coordinates is a realization of a random value such that its distribution is uniform and independent of all other variables. Considering a set of 1,000 random vectors of such type, we obtain the probability distributions of pairwise Euclidian distances and Spearman correlation coefficients, which we have chosen as representatives of the two above-mentioned distance groups (Fig. 5.7).

In this case, regardless of the distance function, unimodal distributions of pairwise distances are obtained. In the next example, a set of vectors is generated by all kinds of permutations of a six-dimensional vector, the coordinates of the vector being pairwise unequal. Under such conditions, the number of the vectors in the set is equal to 720. Fig. 5.8 shows the histograms of the pairwise distances for these vectors obtained with the Euclidian distance and the distances based on the Spearman rank correlation.

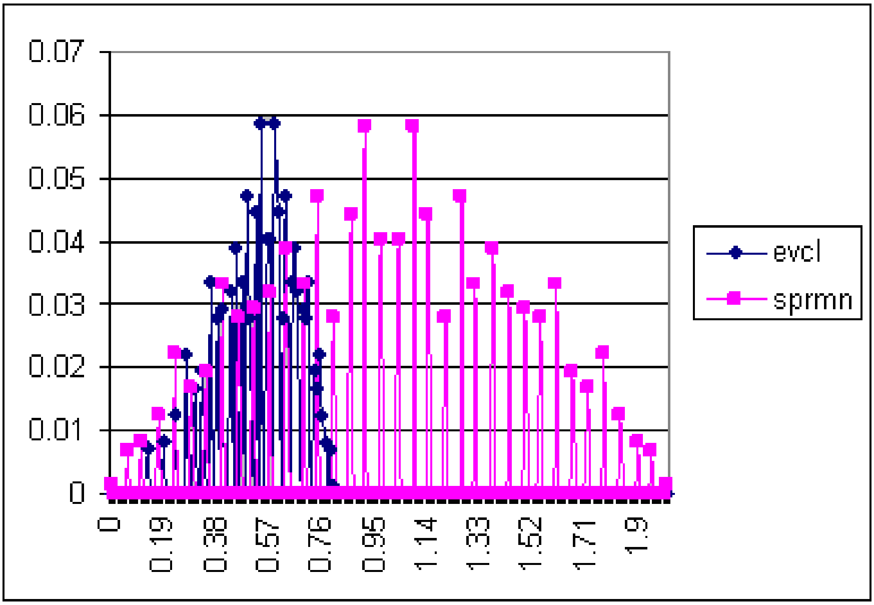


Fig. 5.8 Distribution of the Euclidian distances and of the distances based on Spearman correlation coefficients over the set of 720 non-randomly generated artificial CS. The vector dimension is 6.

It can be seen that the distribution rank with the Spearman correlation distances is more extended than that of the Euclidian distances, but the results are still quite different from those obtained for natural genomes (Fig. 5.5). The fact that the pairwise distance distributions of more or less uniformly simulated CS sets differ from each other means that, in the case of natural genomes, CS sets have specific distributions, which may reflect some peculiarities of the genomic text. It should also be taken into account that the distance distributions are directly simulated due to the direct simulation of the compositional spectra. However, not every distribution on a finite set can be of a CS type, in other words, not every distribution can be viewed as the frequencies of the N-grams which belong to the same sequence.

The determination of the distances between the spectra is the central point of the whole CS approach. A "good distance" is supposed to be in line with the natural perception of spectra proximity. However, there are formal criteria of the distance quality as well. Indeed, it is desirable that parts of the same genome should be close enough to each other, the same being true for the strains of the same species. While this requirement seems to be quite obvious, it is less clear, though no less important, that the distance distribution on a great number of heterogeneous species should be nearly uniform and completely cover the permissible range of distances. The necessity for this condition can be demonstrated by *reductio ad absurdum* in the following way. Let the distance distribution be close to unimodal for a great number of heterogeneous species. This ultimately means that the distance between any pair of species is approximately equal to the mode value and the distribution is described by the dispersion around this value. Such distance scarcely reflects the virtual correlations between genome pairs. For example, the Euclidean distance is affected by the values of N-gram frequencies in a sequence, in other words, by the spectrum form.

Nevertheless, the frequency parameter may bear some "latent" information, which is, actually, "noise" with respect to the spectrum characteristics required for the distance determination. In this regard, the proximity measures such as the Spearman or the Kendall correlation coefficients are the "minimal" ones because they depend only on the differences between the N-gram order in the spectra of the two sequences being compared. We refer to thess proximity measures as the "minimal" ones for it seems impossible to use less data on a spectrum if the distance calculation is based upon all the N-grams of the set W (and not, for example, on any kind of subsampling). Moreover, the meaning of such measures is obvious, namely, if all the N-grams appear in the same place for both spectra, the distance is zero. On the other hand, the more N-grams in one spectrum shift their places with respect to the other spectrum and the greater the magnitude of these shifts, the greater the distance is.

Of course, there may exist other ways of distance definition which we are not aware of at present. Probably, some other distances could provide a more effective processing of the spectrum.

Note that each measure has a certain range of values. For a predefined set of objects, the measure which generates all the pair-wise distances between the objects in the whole range of possible measure values appears to be more sensitive than the measure which gives the pair-wise distances only in a part of this range. Thus, the correlation-based distances between sequences seem to be preferable to all the other distances discussed above. For this reason, it would be sensible to use the d_1 measure, related to the number of coordinate permutations in a vector and based on the rank correlations ρ:

$$\rho = 1 - \frac{6 \sum \Delta_i^2}{n(n^2 - 1)}, \tag{5.18}$$

where $\Delta_i = x_i - y_i$ is the difference between the rank of the corresponding values X_i and Y_i; n is the number of coordinates in each dataset [137]. Let us denote $d_1 = 1 - \rho$, $0 \leq d_1 \leq 2$ [140]. Such distance is in accord with heuristically acceptable understanding of the proximity between two orderings. Therefore, if the distance between two sequences $d_1(S_i, S_j) = 0$, their spectra are identically ordered and we can say that "S_i is compositionally congruent to S_j". In the case of the maximal distance $d_1(S_i, S_j) = 2$, the spectra of the sequences S_i and S_j are ordered in the strictly reverse way.

Though the Spearman and Kendall dissimilarities are defined for any pair of vectors in a multi-dimensional linear space, they are not metrics on this space. Indeed, if the value of the distance d_1 based on the Spearman rank correlation equals zero on a pair of vectors, this just means that their coordinates are identically ordered. A set of vectors with identically ordered coordinates forms a linear space cone since a linear combination of such vectors, with non-negative coefficients, generates a vector which has the same order of coordinates as all the other elements of the cone. From the consideration of a factor-space which identifies all the elements of the cone a finite set is obtained. It is equivalent to the set of permutations which have the corresponding dimension. On this set, the Kendall and Spearman rank correlation do not produce metrics, either. As an example, three vectors a=(1,2,3), b=(2,1,3), and c=(1,3,2) can be considered. It is not difficult to calculate the distances $d_1(a,b)$=6(1+1)/24=0.5, $d_1(a,c)$=6(1+1)/24=0.5, and $d_1(b,c)$=6(1+4+1)/24=1.5 and see that the triangle inequality is not fulfilled.

5.2.4 Compositional Spectra as Non-random and Random Objects

To calculate the compositional spectra of genomes, any set W of N-grams can be used. This set may possess unique features, which can account for the spectra parameters and for the characteristics of the genome sets, based on these parameters. Let us consider two examples of N-gram sets of such type, which, are of considerable interest in themselves, too.

Example 1. All the sub-sequences of fixed length can be taken from a certain genome $\mathfrak{G}$ as N-grams and, according to some specific principle, the set W can be formed using only these N-grams. For example, the set W may consist of a number of the most common N-grams that can be found in the genome $\mathfrak{G}$. Here, the spectra of other genomes and the distances between them are, in a sense, transformed with respect to the reference genome $\mathfrak{G}$.

Example 2. The choice of the set W can be made in the framework of dense spatial packing of spheres or, which is basically the same, of constructing an effective code. First, we will consider this approach for an easier case which corresponds to Definition 3. According to the definition, the occurrence of the N-gram w_i ($w_i \in W$) in the sequence S requires the existence of such a substring y in S that the Hamming distance between y and w_i does not exceed the preset level r. However, it may well happen that the Hamming distance between some other N-gram from W and y does not exceed r, either. There are a number of reasons why this fact impairs the informational value of the spectra. For example, if the described situations are relatively numerous, the values of N-gram occurrences in S may become interdependent, and the spectrum resolution will decrease. Therefore, it would be beneficial to use such set W that the Hamming distance for each N-gram pair of the set is not less than $2r + 1$. This condition means that the distance of each substring y of S is equal to or less than r from not more than one N-gram of W.

The virtual construction of the set W is a separate problem. Generally speaking, a similar problem for the Euclidian space has been known for a long time (see, e.g. [52]). The geometrical interpretation of this problem is dense spatial packing of equal-radii spheres. The solution of the problem in an n-dimensional Euclidian space is mathematically equivalent to that of the major problem of optimal encoding and, as such, is of great applied significance. Moreover, in the cases of 24- and (more than 24)-dimensional spaces, the dense-packing solutions which have been obtained lately have led to important discoveries in mathematics, in particular, in the field of group theory. The main difficulty consists in finding the packing of maximum possible density because even heuristic algorithms of finding reasonably dense packing require serious mathematical efforts. These problems may be substantially simplified if only the most regular configurations, the so-called *latticepacking*, are taken into consideration. Nevertheless, the problem, even in its simplified version, still remains so complex that it was only recently that dense packing of spheres was applied to the practical design of communication systems.

In an n-dimensional Hamming-distance space, a sphere of radius r comprises a join of a set of r-dimensional linear surfaces which intersect in a single point, called the *spherecenter*. Each of the surfaces is parallel to a certain subspace built on the coordinate axes. It is the dense, or optimally dense, packing of such spheres that the problem of constructing the set W is reduced to. Methodologically, this problem is equivalent to the one described above for the Euclidian space. Thus, constructing the set W, which should have certain preset properties and a relatively large volume, may present a difficult task.

In a more general case of N-gram occurrences described by Definition 2 it is required that, for each pair $w_i, w_j \in W$, the sets of minimal elements $\mathfrak{X}_{min}(w_i, S, r)$ and $\mathfrak{X}_{min}(w_j, S, r)$ should have an empty intersection for any sequence S. The construction of such N-gram set is a complicated algorithmical task.

The sets W of the described-above types possess an intrinsic extremal feature, which can account for the obtained result formulated in terms of, e.g., non-intersecting covers. However, generally speaking, these sets are not unique and the question arises as to whether the results depend on the set choice. Due to the above-mentioned technical difficulties which arise in the process of generating such sets, the answer cannot be readily obtained. Therefore, it was proposed [140] to use a random set W (of course, with a certain distribution) for calculating the spectrum and for the subsequent analysis. Since W is a random set, all the functions that involve W have also random values. Even in the case of a uniform distribution of W, the distribution of pair-wise distances between the spectra is of complex nature since the compositional spectra are sequence-dependent. This means that, for a fixed sequence, uniform W distribution does not necessarily result in uniform distribution of the compositional spectra viewed as vectors in the corresponding vector space. Moreover, comparing the distributions of the vectors themselves, one can assess the dissimilarity of two sequences. Thus, the distance between two spectra is a random value and its distribution can be calculated on the basis of the spectrum distribution. In this context, the cluster structure is also random; for example, its stability depends on the reciprocal distribution of pair-wise distances.

In the next chapter, the main technique used for generating an N-gram set W is random uniform sampling from the entire N-gram set. It will be shown later that the spectra parameters can be selected in such a way that the pair-wise distances are almost independent of the random realization of an N-gram set. Nevertheless, the results obtained on the basis of the compositional spectra approach are of statistical nature, so their proper evaluation should be performed. This is of particular importance for the cluster analysis, where minor distance variations can result in significant transformation of the cluster structure.

The following simple models are examples of generating random spectra.

(1) Let N-grams be produced by sequentially adding new letters (of the four-letter alphabet), each letter having equal probability of appearing in the current position. We will refer to such stochastic procedure as *uniformly random* and any random set W of N-grams produced in such a way will also be referred to as a *uniformly random set.*

(2) Every probability vector $\{p_1, p_2, p_3, p_4\}$, $(p_1 + p_2 + p_3 + p_4 = 1,\ p_i \geq 0)$ can be used to generate a set of N-grams, W, so that the frequencies of the letters A, C, G, T will be approximately equal to the corresponding frequencies of the probability vector. Such a random set W will be referred to as *compositionally random.* In these notations, the uniformly random set (0.25, 0.25, 0.25, 0.25) is a compositionally random set.

(3) Markov random set W is created if each N-gram w_i is generated by a predefined Markov chain model, in which the probability of a letter in a given position within an N-gram is a function of k previous letters of the N-gram.

Choosing different W-generating procedures, one can produce various classes of compositional spectra. By analogy with optical spectra, each class of CS can be associated with a certain combination of wave lengths. Similar to illumination of a coloured object with monochromatic light, which makes areas of certain colours more vague, while other areas can be seen more distinctly, the use of CS of different types makes some peculiarities of a sequence obscure, while others become more prominent.

Chapter 6
Application of Compositional Spectra to DNA Sequences

In Chapters 6-8, we bring examples of genome classifications based on different methods of genome presentation. In particular, in this chapter, the compositional spectrum (CS) approach, introduced above, is employed. We choose CS parameters for the application of the method to DNA sequences and consider the main CS properties of the classification obtained. The reduction of the entire N-gram set is quite significant, as compared to the case of corpus reduction, discussed in Chapter 3. The N-grams are chosen randomly under the condition that any letter of the original alphabet can be found in each position of each word with equal probability. Definition 3 (Chapter 5, Section 2) is used for calculating N-gram occurrences in a sequence.

6.1 The Choice of Compositional Spectra Parameters

The approach of using compositional spectra for genome clustering, described in the previous chapter, requires a reasonable choice of three parameters: the length of N-grams (L), the number of N-grams (n) in the set W, and the allowed mismatch (r). The choice of the parameter values is made with regard for the following three constituents:

(1) the problem that is to be solved on the basis of CS; (2) the *a priori* biological requirements imposed on the N-gram parameters, which follow from the problem; (3) informational capacity of the N-gram set.

With regard to (1), the problem that we are going to solve is that of genome classification. Any biologically meaningful genomic classification should reflect the relationships established between ancestors and descendants in the course of evolution. The attempts to evaluate these relationships have been undertaken more than once on various bases, e.g., on the basis of phenotypical traits (see Chapter 2). Since the evolution of phenotypical traits is, ultimately, the evolution of genomes, the philogenetic classification based on genome sequences is of particular interest. The currently adopted genomic phylogeny is built on relatively short conservative (slowly changing) genome

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 87–112, 2010.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

sequences (see Chapter 2). These small-length sequences can be compared almost directly using the alignment technique.

The question arises, however, as to whether whole genomes or their large parts actually reflect evolution. It is hardly possible to compare such sequences by the alignment method (see Chapter 4 for discussion). In this chapter, we are going to describe the comparison of genomes by means of CS technique and building, on this basis, proximity relations between the genomes. The fact that the obtained system of relationships coincides or, at least, is close to the standard genomic phylogeny, suggests that the memory about the origin is distributed along the whole genome. If the obtained relationships do not coincide with the genomic phylogeny, one should try to reveal the source of such clustering, which, in any case, reflects the objective reality.

The problem set up in this way determines the lines along which CS parameters should be searched for, with regard for certain biological considerations, which sends us to item (2). The requirement of the problem is that the N-gram length should be large enough so that the N-gram could not be misidentified as some common universal signal, which does not reflect the specific origin of a genome. For example, words of length 3, which are present in coding sequences, encode 20 aminoacides. Since bacterial genomes almost solely consist of genes, the N-gram frequencies for these genomes are, to a great extent, determined by the frequencies of the aminoacides that the genome encodes. From the philogenetic point of view, this factor is not of major importance, being rather connected with the ecological aspects of the species habitats. Moreover, the protein code impact makes it rather problematic to compare bacterial genomes with those of mammals, where the portion of genes is extremely small. There also exist other universal signals such as two-letter codes of curvature.

The main models of genome evolution (see Chapter 4, Section 4) imply that the comparison of N-grams with the text should allow for mismatches of several letters. It was mentioned in the previous chapter that the allowed mismatch, r. For example, if the original sequence consists of 10 letters T, S = (TTTTTTTTTT), and, in the course of evolution, one mutation (substitution) has occurred, the resulting sequence S' = (TTTTATTTTT). The sequences S and S' have no common words of, say, length 6 and, in the case of no allowed mismatch, they would be in no way similar. However, if just one mismatch of the words is allowed, the sequences appear to be identical. Such result is quite acceptable from the biological point of view since one mutation is not a significant unit of evolution. However, it is not clear what biological considerations could impose restrictions on the value r itself. Indeed, when two protein sequences shown in Fig. 6.1 are compared, there exist only 22 matches over 68 positions. The area marked in the centre of the sequences does not have any matches at all. However, it is quite obvious, that the compared sequences are closely related.

```
DVNLPKFDGFYWCRQIRHESTCPIIFISARAGEMEQIMAIESGADDYITKPFHYDVVMAKIKGQLRR
|||||  |||   |  |  |                      ||||   |||      |     |||
DVNLPGIDGWDLLRRLRERSSARVMMLTGHGRLTDKVRGLDLGADDFMVKPFQFPELLARVRSLLRR
```

Fig. 6.1 An example of the coincidence of two protein sequences with significant mismatch [77].

The informational requirement (3) imposed on the N-gram length and on the volume of the set W, with the allowance (1) made for the problem to be solved, can be reduced to the requirement of sufficient frequency variability of N-gram occurrences in the sequence, the latter requirement being directly connected with the set W informational capacity. In this chapter, the *informational capacity* of an N-gram set is defined as the number of different N-gram frequencies in the sequence. For example, in a natural genomic sequence of length 500,000, at N=3, all the 3-grams have different occurrence frequencies, but the number of all possible 3-grams is only 64. At $N = 9$, the number of different N-grams is as large as 262,144, but the number of different frequencies is only 102 because 9-grams occur very rarely. The optimal informational capacity, i.e., the maximum number of different frequencies, is reached at $N = 5$ or 6 and is equal approximately to 400. Obviously, the requirements for a sufficiently large N-gram length and for a large informational capacity run counter to each other since the genomic text length cannot be, even theoretically, infinite and, therefore, relatively long words very rarely occur in a text. However, considering N-gram occurrences with mismatch r makes it possible to use much greater values of N.

For further condideration, we need to introduce some formal definitions. Namely, let I designate a set of positive integers, $I = \{i_1,i_2,\dots,i_r\}$, such that all of them are pairwise different and do not exceed the value of $N + |I|$. Next, we define a $distributed(N, I) - gram$ as a string of $N + |I|$ symbols, where a conventional symbol of a "gap" occurs in fixed positions $i_1,i_2,\dots,i_p$, while all other symbols belong to the original alphabet. For example, the strings $TCGG$ and $TC - GG$ are a 4-gram and a (4,I)-gram (I=\{3\}), respectively. We will say that a (N,I)-gram occurs in the sequence if there is a match in all the (N,I)-gram positions occupied by the original alphabet symbols, while the "gap" positions are disregarded (Fig. 6.2).

```
S: ...ATTCGGTACTTG...           S: ...TCGTCAGGCGA
        TCGG                               TC - GG

        (A)                                (B)
```

Fig. 6.2 The scheme of N-gram (A) and (N,I)-gram (B) occurrences in the sequence S.

Obviously, the distributed (N,I)-gram with the empty set I ($I = \emptyset$) is a usual N-gram ((N, $\emptyset$)-gram $\equiv$ N-gram) and the number of all possible (N,I)-grams is independent of the set I and, with the four-letter alphabet, is equal to 2(2N) . Let us designate the average number of (N, I)-gram occurrences in the sequence S as $M(N, I, S)$. This value is readily expressed as: $M(N, I, S)$=(|S|-(N+|I|))/2(2N). If the value of |I| is small enough (say, does not exceed N), M(N,I,S) is virtually independent of |I| and, for example, at |S|=500000 and N=6, N=7, N=8, and N=9, M(N,I,S) is equal to 122, 30, 7, and 2, respectively.

The distributed (N,I)-gram provides the means to describe all the strings which can be constructed in the case of mismatch between an N-gram and the sequence being allowed. For example, let, as usual, the mismatch be r. Then, for each N-gram, the question of its occurrence in the sequence S with mismatch r is reduced to the question of the ideal-match occurrences of all distributed ($N - r$, I, S)-grams at $|I| = r$.

Now we can formulate the following empirical estimation H of the mean N-gram occurrence with the mismatch r:

$$H = C_N^r \sum_{|I|=r} M(N - r, I, S).$$

Let, for example, N=10 and $r = 1$. Then, each N-gram generates 10 generalized (9,{i})-grams. Since, at N=9, the average number of (N+1)-gram occurrences is equal to 2, the average number of 10-gram occurrences with one allowed mismatch may be estimated as 20. With two allowed mismatches, the number of possible generalized (8,{i,j})-grams is equal to 45 so that the corresponding expected number of 10-gram occurrences may be estimated as 7*45=215, where 7 is the average number of 8-gram occurrences. Finally, with three allowed mismatches, the average number of (7,{i,j,t})-gram occurrences is equal to $30 * 120 = 3600$. The two latter values are sufficient for producing a relatively large informational capacity of the 10-gram set. Using the above estimation of H, we can calculate the preferable values of r at different N values (Table 6.1) which should, presumably, give the required informational capacity (the values of H are set to lie in the range 3,000-10,000).

Theoretically, the chosen value of N may lie in the range specified by Table 6.1 or be even larger; however, there are some additional considerations which allow to restrict the choice. For example, the N-gram length equal to 10 appears to be informative enough for the problem of reconstructing the

Table 6.1 Variation of the mismatch r with the N-gram length at the fixed range (3,000-10,000) of average N-gram occurrences.

N	10	12	15	20	25
r	3	4	6	8	11

whole sequence from its fragments [211]. The informational capacity of the whole N-gram vocabulary was also defined as a certain relationship between the characteristics of two consequent whole N-gram and (N-1)-gram vocabularies [219]. The authors showed, as an example, that the *Borrelia burgdorferi* genome informational capacity has its optimum at N=11-12. It also appears that a too long N-gram does not have sufficient "flexibility" to appropriately reflect the possible genome "fragmentary character", while a too large value of the allowed mismatch may lead to artifacts. Therefore, for the sake of definiteness, we choose the values of N=10 and r=2,3, although the values of N=11 or 13 could well be considered, too.

Obviously, *a priori* checking the adequateness of the chosen values of CS parameters and of the set W volume is impossible and the values should be approved on natural genome texts. With that end in view, throughout this chapter, we use a database that comprises the genomes of as many as 37 species which belong to all the three Kingdoms - Eukaryota, Eubacteria, and Archaea.

The sequences represent large stretches sampled from the data on the genomes; for each species, two different target 200-500 Kb sequences (referred to as A and B) are produced. In a few cases, when the available material is only sufficient to build one target sequence (A), the B sequence is taken equal to A in order to maintain the same structure of the algorithms. In the case of the *Homo Sapience* genome, 11 fragments of different chromosomes are considered (see The List of Depicted Genomic Sequences).

From the fact that the CS method employs a random set of N-grams, W, it immediately follows that the compositional spectra $F(W, S)$ and $F(W, S$') of the sequences S and S', respectively, are also random; hence the distance between the spectra of the sequences S and S' is also a random value. Let us analyze the distribution of the random variable d_1 (see Chapter 5, Section 5.2.3 for the definition of d_1) produced by a uniformly random set W. The main empiric result is that, in this case, the measure $d_1(S, S')$ is statistically stable when S and S' are genomic sequences. Indeed, the distribution of $d_1(S, S')$ appears to be close to normal and its standard deviation decreases with the increase of the number of words, n, in the set W. For 100 uniformly random sets and for every pair of species, i and j (i, j belong to the collection of 38 species, $i \neq j$), σ_{ij} is the standard deviation of the distances d_1 over all tested sets W. Then, the averaged standard deviation for all possible pairs i, j,

$$\bar{\sigma} = \frac{\sum_{i \neq j} \sigma_{ij}}{\mathfrak{d}}$$

(where $\mathfrak{d}$ is the number of different pairs of species), can be considered as an indicator of the robustness of the measure d_1 for given N, n, r.

The standard deviations of the distances d_1 for $N = 10$ and $r = 2$ are shown in Table 8.2 and in Fig. 6.3 for different n.

Table 6.2 Deviation of the mean standard distance (d_1) between two genomic sequences as a function of the number of words, n, in the set W.

n	25	50	100	200	300
$\bar{\sigma}$	0.15	0.1	0.09	0.05	0.05

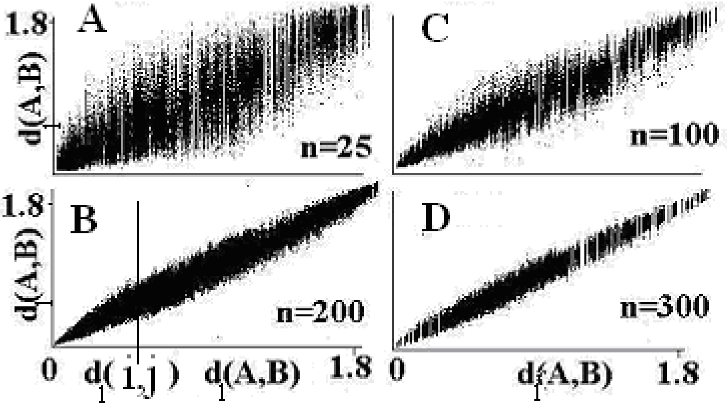


Fig. 6.3 The effect of the vocabulary size (n) on CS-distances. X-axis: the values of all possible pair-wise CS-distances for a particular N-gram set W_o. Y-axis: for each distance $d_1(i,j)$, its values, calculated for 100 random implementations of the set W, are plotted along the vertical line which passes through the point $d_1(i,j)$ on the X axis. As an example, such vertical line is drawn in (**B**).

From the data shown in Fig. 6.3, it can be seen that the variation of CS decreases with increasing the W size for any pair of sequences. Thus, the consistency of the genome comparison by means of compositional spectra increases with n up to a certain point of saturation, which corresponds to n ~200. One can see that the size n=200 is, indeed, a reasonable asymptotic choice. Similar results can be obtained for compositionally random variables W and Markovian random sets. Thus, the distances between sequences estimated on the basis of any random set W are sufficiently consistent.

To sum up, it should be emphasized that there is no unique criterion for the optimal parameters of a compositional spectrum. However, the above considerations show that these parameters can be chosen within certain limits of the spectrum "validity". In what follows, we will use the parameter values of N=10, n=200, and r=2.

The relatively small value of the W set volume with respect to all possible N-grams is compensated by the fact that, due to the allowed mismatch, each N-gram of the set generates a relatively large set of neighbouring N-grams. For example, at N=10, the number of N-grams which differ from a particular N-gram by exactly three mismatches is equal to $120 * 81 = 9720$, while at

Table 6.3 Characteristics of sequence covering for L=10, n=200, and different values of r.

r	min %	max %	mean %	75%
1	0.17	0.76	0.45	0.25-0.5
2	3.1	13.1	7.2	3.4 – 9.0
3	31.0	87.0	57.0	45.0-60.0

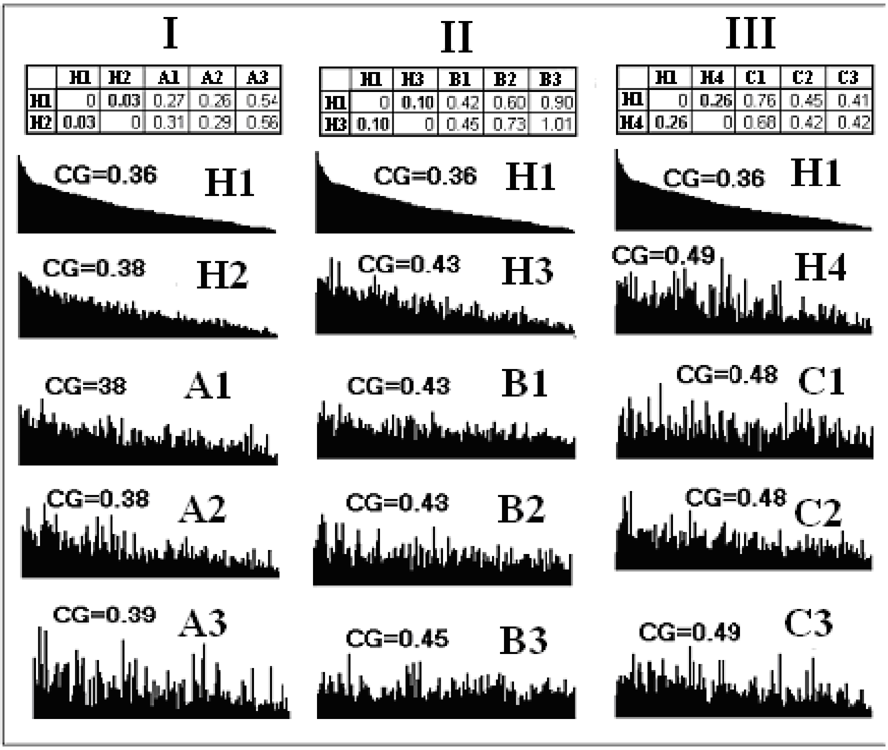
n=200, the total number of such "close" N-grams is 2,000,000, the value being larger than the number of different 10-grams. This result is, obviously, produced by overlapping of neighbouring N-grams. In order to conceive the volumes of the W set and of the neighbouring N-gram sets at the allowed values of mismatch r, let us calculate the *set covering*, i.e., the total number of the sequence letters which occur, in at least one N-gram which belongs to this sequence.

For the N-gram length N=10 and n=200, the sequence covering characteristics for three values of r (r=1, 2, 3) are presented in Table 6.3. In the second column of the table, "min %" means the minimum percent of covering by a random set of words for all sequences from a certain test set. Similarly, "max %" is the maximum percent of covering for the same set of sequences, while "mean %" is the mean value of covering for all sequences. However, for the majority (75%) of the analyzed sequences, we may find a narrower covering interval (min % – max %), which is shown in the last column of the table.

6.2 The Effect of *GC* Content on Compositional Spectra

Obviously, the variations in a particular N-gram occurrence in different genomes may be accounted for by variations in the frequencies of the alphabet letters in each of these genomes. It is the $A+T$ or $C+G$ frequencies that are usually considered since, acording to the Chargaff's rule 1 (see Chapter 1, Section), the letter A frequency is equal to that of letter T, the same being true for the letters G and C. As the sum of the A+T and C+G frequencies equals 1, the frequency of only one pair of letters, traditionally C+G, is usually specified. One could suppose that the observed variation of CS-distances is related to the G+C content in different genomes. The results of the two tests described below show that the effect of $G + C$ content is too small to account for the observed variations of CS-distances for different species.

Test 1. The results of this test are presented in Fig. 6.4 in the form of a table with three columns and six rows. The arrangement of the top and the bottom rows is different from that of the other rows. In the first (top) row the spectrum related to a fragment of the *H. Sapiens* chromosome 1 is repeated three times to appear in all three columns because this spectrum



I

	H1	H2	A1	A2	A3
H1	0	0.03	0.27	0.26	0.54
H2	0.03	0	0.31	0.29	0.56

II

	H1	H3	B1	B2	B3
H1	0	0.10	0.42	0.60	0.90
H3	0.10	0	0.45	0.73	1.01

III

	H1	H4	C1	C2	C3
H1	0	0.26	0.76	0.45	0.41
H4	0.26	0	0.68	0.42	0.42

Fig. 6.4 H1-H4: Human contigs from different chromosomes; **A1-A3**, **B1-B3**, **C1-C3**: contigs from other species. **I**: All contigs have approximately the same $G+C$ content; **II**: $G+C$ content of H3 and B1-B3 is approximately the same and higher than that of H1. **III**: G+C content of H4 and C1-C3 is approximately the same and still higher than that of H3 and B1-B3. Values of CS-distances between the Human contigs and all the other contigs are given in the tables at the top of each column.

serves as a reference ranking. The second row comprises spectra related to three different human chromosomes. The next three rows represent spectra related to various organisms, which were selected in such a way that their $G{+}C$ contents are approximately equal to those of the corresponding human genome fragments in the second row. The tables at the top of each column show the distances d_1 for two human genomes and other genomes from the same column. It should be noted that the distances between all the pairs of human genome fragments are always smaller than the distance between a human genome fragment and an unrelated sequence although the $G+C$ content of the latter two sequences is the same.

Test 2. This test shows that the closeness of genomes depends rather on the sequence composition than on the $G{+}C$ content similarity. We will show that it is possible the change significantly the distances between sequences by merely changing the sequence composition, while the *G+C* content and, which is more, the *G+C* positions stay unchanged.

Let each species in the database be presented by two genome fragments, *A* and *B*. For the purpose of this test, each *B* fragment may be randomized in a few different ways. Namely, in the process of *GC-reshuffling*, *G* and *C* letters may be transposed randomly, while the positions of *G* and *C* in the sequences are preserved. The same procedure can be performed with *A* and *T* letters (*AT-reshuffling*). A complete random mixing the letters, regardless of their positions, can also be performed. The distributions of such "pseudo-intragenomic" distances between the original *A* segment and the reshuffled *B* sequence are presented at the top of Fig. 6.5 (row I). From the comparison of this figure with Fig. 5.6 in Chapter 5, it can be inferred that each of the above reshufflings substantially increases the original CS-distances between the fragments. For example, while the original intragenomic distances range from 0 to 0.2 (Fig. 5.6A, Chapter 5), the pseudo-intragenomic distances fall in the ranges 0.04-0.66 and 0.04-0.9 in the cases of *GC*- and (*GC+AT*)-reshuffling, respectively.

The dependence of the distance d_1 on *G+C* content at the intergenomic level is assessed using the model of contig *B*-reshuffling described above, with subsequent calculation of the pair-wise distances between the contigs. The distributions of such "pseudo-intergenomic" distances are presented in Fig. 6.5, row II. It can be seen that the shapes of the histograms are notably different from the original shape (Fig. 5.6, Chapter 5). The similarity of *G+C* contents results in smaller distances, which corresponds to the peak at the beginning of the histograms, while for significantly different G+C contents the distances are large (the peak at the end of the histograms). In contrast to this, in the original case, without reshuffling, the majority of the distances are located in the central part of the histogram, which once again demonstrates that the primary structure of a genomic sequence reduces the effect of *G+C* content on the distances.

It should be pointed out that the differences among the compositional spectra of the sequences compared above cannot be accounted for merely by variations in *G+C* content. For example, if the difference between the *G+C* content of two fragments from the same genome does not exceed 10%, the distance between the fragments is almost independent of the G+C content differences.

However, this is, definitely, not the case for intergenomic distances for certain pairs of species. In order to assess the relationship between $G + C$ content and CS-distances, the distance D_{C+G} is introduced as the absolute value of the difference between the species $G + C$ content. The results presented in Fig. 6.6 show that when the D_{C+G}-distance is less than 20-25%, there is almost no relationship between D_{C+G} and distances d_1, whereas higher differences in G+C content have a profound effect on d_1.

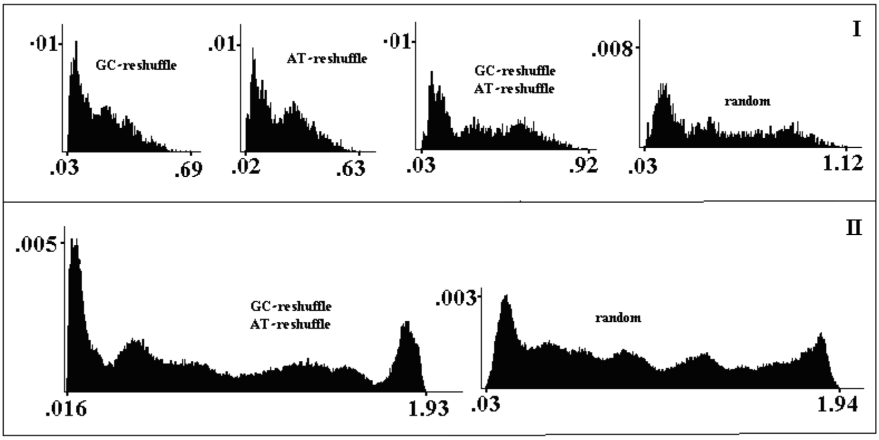


Fig. 6.5 Distributions of pseudo intra- and intergenomic CS-distances based on the Spearman rank correlation. Row **I**: distributions of pseudo intragenomic distances between the original fragment A and the reshuffled fragment B. Row **II**: distributions of pseudo intergenomic distances between the reshuffled fragments B.

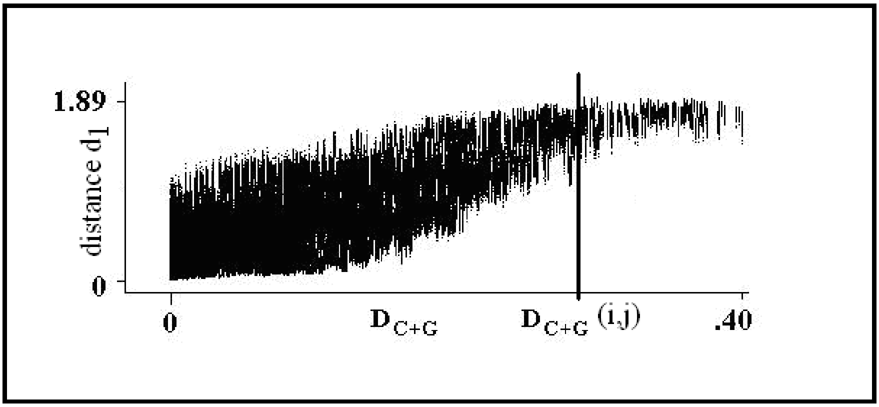


Fig. 6.6 Effect of $C+G$ content on distance d_1 between genomic sequences. Points $(D_{C+G}(i,j), d_1(i,j))$ are presented for each of the two contigs, i and j. $D_{C+G}(\,i,j)$ is the absolute value of the $C+G$ content difference between the contigs i, j; $d_1(i,j)$ is calculated for the spectra built on 100 randomly chosen sets W.

6.3 Associated Spectra and Projections of Sequences

6.3.1 Derivative Spectra

For any chosen set W, a related set W^* of reverted complementary N-grams can be constructed. This means that each N-gram should be

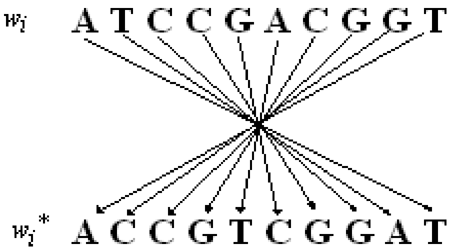


Fig. 6.7 The scheme of constructing the reverted complementary N-gram (w_i^*) on the basis of the N-gram w_i.

inverted, i.e., written from left to write, each element of (A,T)- and (C,G)-pairs being replaced by the other element of the same pair. For example, if $w_i = ATCCGACGGT$, then $w_i^* = ACCGTCGGAT$ (Fig. 6.7).

Thus, for each sequence S, two spectra can be calculated - $F(W,S)$ and $F(W^*,S)$. If the N-grams of the set W^* are ordered in the same way as those of the set W, i.e., on the basis of the intrinsic correspondence between the N-grams w_i and w_i^*, the spectra can be compared directly. It is worthwhile noting that the spectra $F(W, S)$ and $F(W^*,S)$ are, as a rule, highly correlated.

One can also produce other derivatives of the chosen set W, for example, a reverted, but not complementary, sequence w_i^{**} (i.e., $w^{**} = TGGCAGCCTA$ for $w = ATCCGACGGT$). It can be shown that the resulting set W^{**} also produces informative spectra; however, in contrast to the foregoing situation of the $F(W, S)$ and $F(W^*, S)$ spectra being highly similar, the $F(W, S)$ and $F(W^{**}, S)$ spectra do not appear to be correlated at all. This indicates that, in addition to some general species-specific structure in the genome organization revealed by any set of words, some additional palindrome-like patterns may also exist [90].

6.3.2 Two-Letter Alphabet

In Chapter 1, it was shown that the DNA sequence obeys certain empirical rules, having, undoubtedly, by biological reasons, which are not quite clear yet. For example, in the case of sufficiently large fragments, the frequencies of A and T letters (as well as those of C and G letters) are approximately equal. On the other hand, the letters of the DNA alphabet are, actually, four nucleotides, which can be identified on the basis of their physico-chemical properties. Purine derivatives, A and G molecules, have similar chemical structure, while T and C molecules, being pyrimidine derivatives, are also structurally similar. Identification of A and G as puines and T and C as pyrimidines results in a two-letter purine/pyrimidine alphabet, which is designated here as (R,Y). This alphabet is interesting, in particular, in that,

R and Y letters almost always have the same frequencies (equal, obviously, to 0.5).

With the two-letter (R,Y) alphabet, it seems sensible to choose N=20 so that the number of possible N-grams would be the same as with the four-letter alphabet for N=10. Similar to the four-letter case, the conditions of n=200 and r=2 are chosen for the two-letter alphabet. The similarity of the parameters in the cases of the four- and the two-letter alphabets makes it possible to compare the species classifications obtained by the compositional spectra with the two types of alphabets. To produce a set of words, W, a random generator can be employed under the assumption that each of the four (or two) symbols appears at any current position with equal probability.

6.4 Different Genome Clusterings Obtained with the Four- and the Two-Letter Alphabets

In this section, we describe the application of the CS technique to the problem of genome classification. The results obtained in the form of dendrograms can be directly compared to the phylogenetic trees described in Chapter 2.

6.4.1 Two Different Classifications of Organisms

A common technique of building statistically significant phylogenetic trees is the method of constructing a *consensus tree*. This method is based on building a certain set of phylogenetic trees, which is generalized in the form of a consensus tree [2, 180]. In this chapter, two alphabets, a four-letter (A,T,C,G) and a two-letter (R,Y) alphabet, are employed for CS calculation. A hundred sets of random words are generated and the spectra resulting from each set are used for calculating a matrix of pairwise distances between the species under consideration. From the set of phylogenetic trees, the consensus tree is, finally, obtained.

Fig. 6.8 shows a dendrogram obtained with the four-letter alphabet. This structure differs considerably from the standard three-kingdom scheme (se Chapter 2). It has proved convenient to analyse the dendrogram, distinguishing three clusters on it (see Fig. 6.8). Cluster I contains all the considered sequences from the human, mouse, and *A. thaliana* mitochondrial genomes as well as the sequences from several Eukarya and thermophylic Archaea. Cluster II contains a number of eukaryotes and prokaryotes, while cluster III contains bacteria (both Eubacteria and Archaea) and a free-living eukaryote *Leishmania*. We can see that, in the obtained clustering, Eukarya sequences appear in two out of three clusters, whereas Archaea can be found in all the three clusters. It can be supposed that the key factor in the classification based on the (A,T,G,C) alphabet is related to ecology: two ecological parameters, temperature and oxygen, almost perfectly account for the clustering peculiarities. For example, the composition of cluster I, which includes Eukarya, ,thermophilic bacteria, and a part of the studied Archaea, may have resulted from two interdependent evolutionary processes: the evolutionary divergence and the superimposed ecological convergence of the genomes, albeit another process, horizontal transfer, cannot be excluded as a contributing factor. The three clusters (I, II, III) and such partition structure shown in Fig. 6.8) will be referred to as the *main pattern* (MP; see Table 6.4) and the *mixed-structure* (or the *m-structure*), respectively (for details, see [139]).

Table 6.4 Mixed structure obtained on the basis of CS distances. **Cluster I**: all considered sequences from the human, mouse, and *A. thaliana* mitochondrial genomes and sequences from several *Eukarya* and thermophylic *Archaea*. **Cluster II**: a number of eukaryotes and prokaryotes. **Cluster III:** bacteria (both *Eubacteria* and *Archaea*) and the free-living *Leishmania*.

Cluster	Eubacteria	Archaebacteria	Eukarya
I	*T. maritima*	*P. horikoshii*	*H. sapiens* chr. X,Y
	A. aeolicus	*P. abyssi*	*M. musculus*
		A. fulgidus	*A. thaliana mitochondria*
		M.thermoautotrophicum	
II	*R. prowazekii*	*M. jannaschii*	*D. melanogaster*
	B. subtilis	*S. solfataricus*	*C. elegans*
	M. genitalium		*A. thaliana*
	M. pneumoniane		*S. cerevisiae*
	E. faecalis		
	H. pylori		
	H. influenzae		
	S. pyogenes		
	C. acetbutylicum		
	B. burgdorferi		
	C. jejuni		
	Synechocystis sp.		
III	*T. palladium*	*A. pernix*	*L. major*
	E.coli	*Halobacterium sp. NRC-1*	
	M. tuberculosis		
	N. gonorhhoneae		
	N. meningitides		
	D. radiodurans		
	A. actinomycetemcomitans		
	P. aeruginosa		
	T. thermophilus		

From the data presented in Table 6.4 it can be inferred that, if the CS-distances are obtained with the four-letter alphabet, the main-pattern subdivision displays the essential features of clustering, while the traditional three-kingdom scheme (Eukarya, Eubacteria, and Archaea) does not fit the clustering criteria.

The dendrogram which is based on the compositional spectra obtained with the two-letter alphabet (R,Y) is presented in Fig. 6.9 [142]. Similar to the case of the four-letter alphabet, in this tree, it is also possible to distinguish the main pattern - the three main clusters (Table 6.5). The obtained structure displays a much better similarity to the standard three-kingdom scheme than the corresponding dendrogram obtained with the four-letter alphabet (see Fig. 6.9). Indeed, in Fig. 6.9, cluster I includes nearly all the considered Eukarya (with the exception of *S. cerevisiae* and *Leishmania major*). Cluster II includes all the considered Archaea,, all thermophylic Eubacteria, and three other bacteria. Cluster III consists of bacterial species. This three-cluster partition will be referred to as the *pattern of two-letter structuring.*

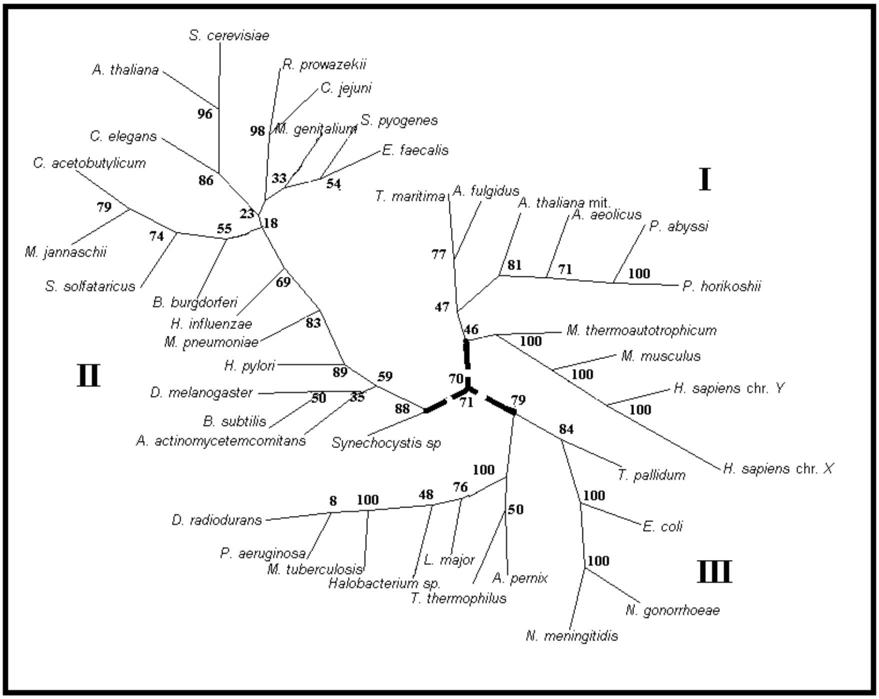


Fig. 6.8 Dendrogram based on the four-letter compositional spectra.**I-III** are the three main clusters; the numbers near the nodes are the robustness of the corresponding fission based on the bootstrap analysis.

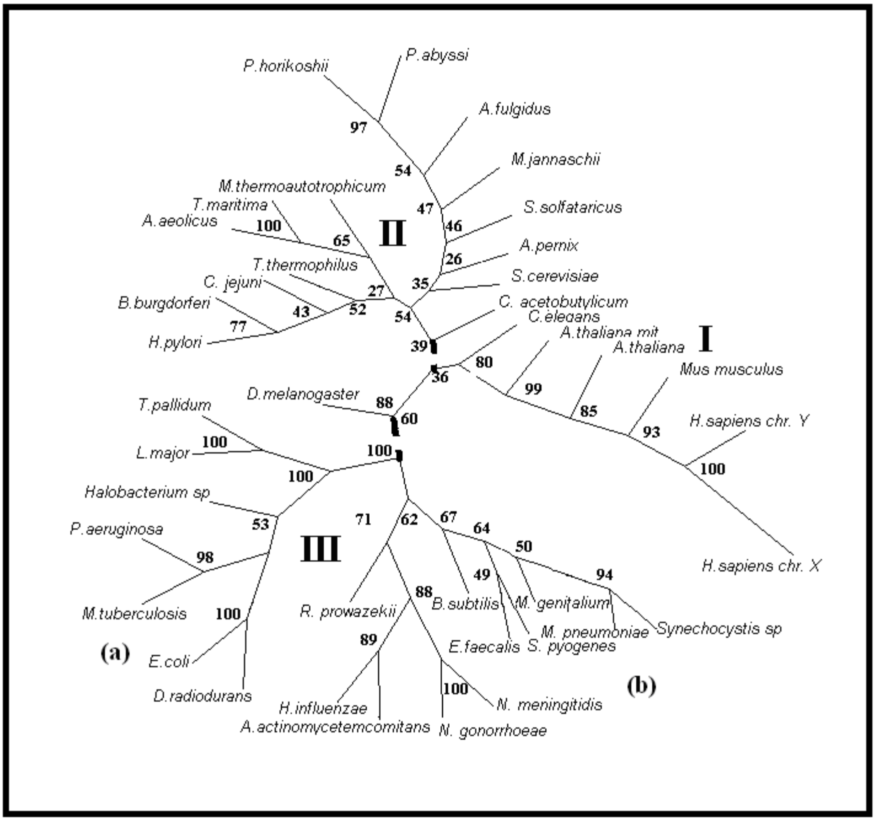


Fig. 6.9 Dendrogram based on the two-letter compositional spectra. **I-III** - the three main clusters obtained; the numbers near the nodes are the robustness of the corresponding fission based on the bootstrap analysis.

The comparison with the traditional three-kingdom scheme considered above is based on distinguishing three main clusters in each of the above two dendrograms (Table 6.5).

However, distinguishing these clusters as sub-clusters of the dendrogram (which is the main cluster) has been performed empirically. The point is that the traditional methods of evaluating dendrograms are, actually, hierarchical (agglomerative) clustering methods. Such methods consist of subsequent steps, each one uniting the two previously obtained clusters into a single cluster. The process is finished after all the elements have been united into one cluster, which is just the hierarchical tree. In contrast to the "partition" clustering methods, the standard scheme of this process does not allow for the definition of sub-clusters. Therefore, in the hierarchical clustering method, we define a sub-cluster as a cluster that appears at some intermediate step of the process. From the structure of the resulting hierarchical tree, it is

Table 6.5 Cluster **I:** almost all considered *Eukarya* (with the exception of *S. cerevisiae* and *Leishmania major*). Cluster **II:** all considered *Archea* (except for *Halobacterium*), all thermophylic *Eubacteria*, and three other bacteria. Cluster **III** consists of bacterial species.

Cluster	Eubacteria	Archaebacteria	Eukarya
I			*H. sapiens* chr. X,Y
			M. musculus
			A. thaliana mitochondria
			A. thaliana
			D. melanogaster
			C. elegans
II	*T. thermophilus*	*M. jannaschii*	*S. cerevisiae*
	T. maritima	*S. solfataricus*	
	A. aeolicus	*M. thermoautotrophicum*	
	H. pylori	*P. horikoshi*	
	B. burgdorferi	*P. abyssi*	
	C. acetbutylicum	*A. fulgidus*	
	C. jejuni	*A. pernix*	
	Synechocystis sp		
III	*T. palladium*	*Halobacterium sp. NRC-1*	*L. major*
	E.coli		
	M. tuberculosis		
	N. gonorhhoneae		
	N. meningitides		
	D. radiodurans		
	A. actinomycetemcomitans		
	P. aeruginosa		
	H. influenzae		
	S. pyogenes		
	R. prowazekii		
	B. subtilis		
	M. genitalium		
	M. pneumoniane		
	E. faecalis		

impossible to determine whether a certain cluster appeared as an isolated object at a particular step of the uniting process or it was constituted as an isolated object from many other tree vertices. At the same time, observing the agglomeration process, one can determine which clusters are formed at each algorithm step. This is not possible, however, in the case of the consensus tree. In what follows, we will describe a procedure which allows to answer the question as to whether a certain subset of the consensus tree represents a sub-cluster. The algorithm employs the same datasets as those that were used for building the consensus tree.

6.5 General Procedure of Cluster Verification

In order to make large-scale comparisons of different species, it is necessary, first of all, to generate 100 sets of words, W_i (i=1,. . . ,100). Calculation of the spectra for all the DNA sequences under consideration for each set W_i results in a matrix D_i of pairwise distances between the sequences Next, by applying the described above methods of clustering to each matrix D_i, the hierarchic structures of embedded clusters are obtained. Such structures appear automatically when an agglomerative technique is applied. The idea of this technique is starting with the points as individual clusters and, at each step, merging the two most similar (the closest) clusters.

First, a certain pre-selected group of species, which are supposed to belong to the same cluster, is marked. Second, the system of clusters is built step-by-step by applying, e.g., the WPGMA method. By definition, at the initial stage, each species is considered to be an elementary cluster. At each subsequent stage, the algorithm joins the two nearest clusters (i.e., groups of species), which is a regular way of agglomerative algorithm functioning. However, in the considered test, the process of agglomeration is stopped at the stage where no less than 75% of the marked group of species is united in the same cluster, which is denoted as a U-cluster.

It should be noted that the U-cluster is the minimal (by volume) set in the WPGMA integration process that includes no less than 75% of elements of the selected group of species. It can be readily seen from the cluster structure that, if the marked species are directly joined in the course of the clustering process, the U-cluster will, probably, contain: (i) more than the required 75% of the marked species, but (ii) comparatively few species which do not belong to this group. Moreover, if, in the process of cluster merging, the marked species join the non-marked ones, the $U-$cluster will contain each of the marked species, together with a number of non-marked ones. Thus, the goal of the test is to evaluate the number of such non-marked species that are absorbed by the U-cluster during the merging process required for uniting "almost all" marked species. The above two possibilities of cluster merging are demonstrated in Fig. 6.10.

Thus, for a fixed set of the marked elements, a hundred of U-clusters are calculated. By definition, each of them contains no less than 75% of the marked elements as well as some others. Averaging all these clusters, we can evaluate the "real" cluster, which is, in a sense, "associated" with the set of the marked elements. The procedure is illustrated by the following examples.

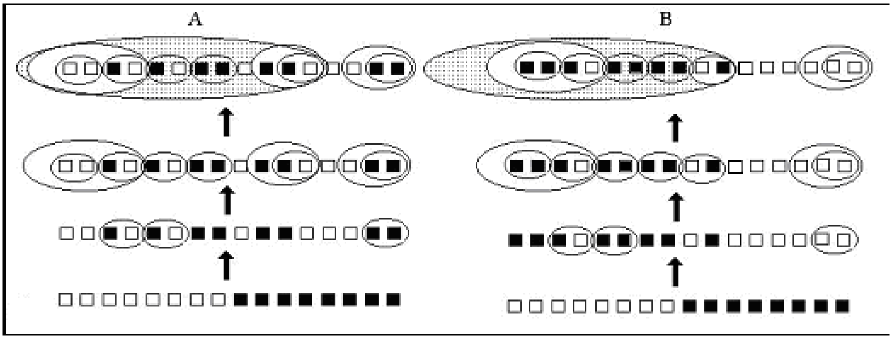


Fig. 6.10 Schematic presentation of U-cluster formation. Marked and non-marked elements are in black and in white, respectively. **A**: Marked elements are poorly bound (many non-marked elements are required for binding). **B**: Marked elements are well bound to each other. In both cases, the process of cluster merging is terminated as soon as the first U-cluster (shaded) appears (harboring ≥75% of the marked elements).

Consider seven elements, a_1, a_2, a_3, a_4, c_1, c_2, c_3, the first four of them being marked. Suppose that the following U-clusters were formed as a result of three clusterizations: $\{a_1, a_2, a_3, c_1\}$, $\{a_1, a_3, a_4, \mathrm{c}_2\}$, and $\{a_2, a_3, a_4, c_3\}$. Obviously, each cluster includes 75% elements of the marked set and one non-marked element. Averaging over all the three clusters shows that the element a_3 occurs in each cluster (the occurrence is 100%), the elements a_1, a_2, a_4 occur in two out of three clusters (the occurrence is 66%), while the elements c_1, c_2, c_3 occur in only 33% of all clusters. This result implies that the occurrence of the latter three elements in each cluster is essentially random, while the set $\{a_1, a_2, a_3, a_4\}$ is, actually, a cluster though its elements may not occur in some cluster samplings (also at random) .

Now let the result of three clusterizations be: $\{a_1, a_2, a_3, c_1\}$, $\{a_1, a_3, a_4, c_1, c_2\}$, and $\{a_2, a_3, c_1, c_3\}$. In this case, the occurrences of the elements are the following: the element a_3 occurs in all the clusters (the occurrence is 100%), the elements a_1, a_2, c_1 occur in two clusters out of three (the occurrence is 66%), while the elements a_4, c_2, c_3 occur in only 33% of all the clusters. This result shows that, in this case, the elements $\{a_1, a_2, a_3, a_4\}$ do not constitute a cluster.

Note that the chosen threshold value of 75% for the clustering procedure termination is an arbitrary one. However, its level should be higher than 50% so that the U-cluster could absorb the majority of marked elements. On the other hand, it should be less than 100% so that the associated non-marked elements could be filtered out. Below, the described test is applied to the evaluation of the cluster stability.

6.5.1 Mixed-Structure and Its Stability

As an example, let us apply the described-above method to testing the cluster structure of the species listed in Table 6.5. Let us mark all the elements of cluster I (except for the mouse and the *A. thaliana* mitochondrial genomes), which includes, mainly, human and thermophylic bacteria genomes (see Table 6.5). For each set of words W_i (i=1,...,100), the U-cluster is obtained and denoted as U_i in order to emphasize its dependence on the set W_i. Next, the proportion of each species occurrences in the set of U_i-clusters is calculated (Fig. 6.11A).

According to the results presented in Fig. 6.11A, all human sequences occur in the U_i-clusters for each set W_i. It should be noted that the same is true for the mouse genome though it is not marked as a member of the "selected group". Other marked sequences from cluster I appear in more than 80% of the U_i-clusters. Remarkably, the *A. thaliana* mitochondrial genome, which is not marked, also belongs to nearly 70% of U_i-clusters. A certain number of non-marked species that do not belong to cluster I also occur in different U_i-clusters. However, in this case, the occurrence of each species does not exceed 20% of all the U_i-clusters. The results of the same test conducted for two other clusters, II and III, is shown in Figs. 6.12A and 6.13A, respectively. The stability of these clusters, as manifested by the U_i-cluster pattern, is similar to that found for cluster I. Thus, it can be concluded that the m-structure, defined in Section 6.4.1, actually reflects some robust (objective) relationships among the considered "across-life" species.

In order to illustrate the results of the test that would be obtained if the assumed cluster structure did not actually exist, let us consider the following two examples.

6.5.2 Effect of Mis-anchoring

In the first example, let us, again, consider cluster I, in which the four thermophilic Archaea are substituted with other four Archaea from clusters II and III. In other words, in addition to human sequences, four new Archaea genomes, which actually should not belong to cluster I, are now "marked". The application of the described-above test to such mixed group gives the result shown in Fig. 6.11B. In this case, in contrast to the previous distinct clustering pattern (Fig. 6.11A), a significant number of non-marked species occurs in each U_i-cluster with nearly the same frequency as those of the marked species (Fig. 6.11B). As before, the same test was conducted for clusters II and III and similar results were obtained (Figs. 6.12B and 6.13B).

6.5.3 GC-Permutation Test

The second example is aimed at testing the effect of $G + C$ content on the m-structure formation and on the patterns that are formed in the U-clusters.

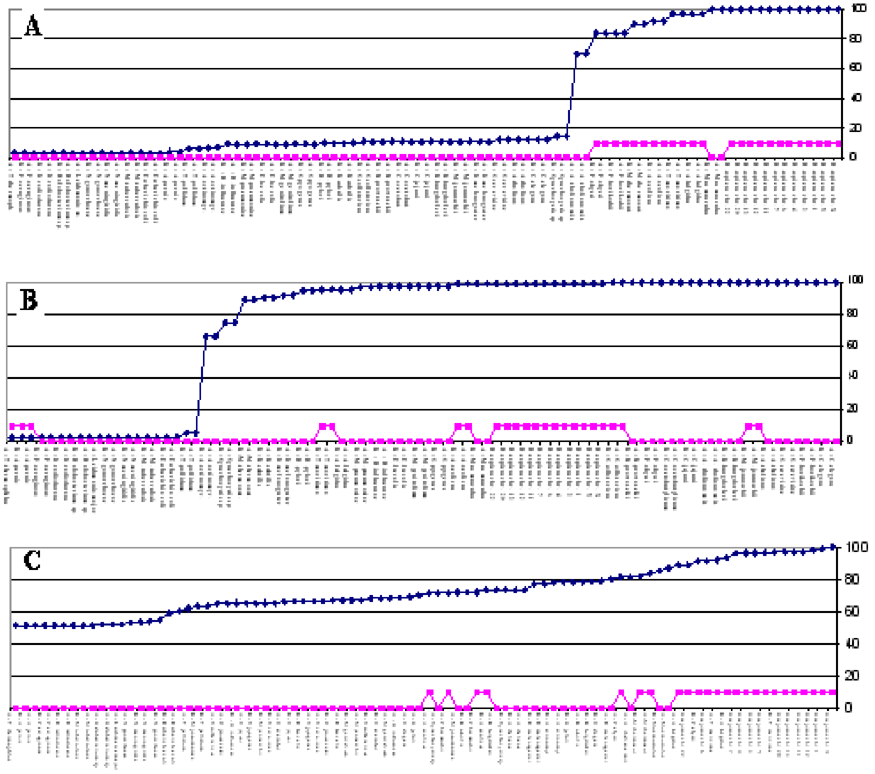


Fig. 6.11 Frequencies of the species occurrences in cluster I. ***X*** axis: names of the species in decreasing order with respect to their occurrences in the clusters. In each plot, the upper curve corresponds to the percentage of occurrences; the lower curve indicates the "selected species group" (y=10 and 0 correspond to the marked and non-marked species, respectively). ***Y*** axis: percentage of the species occurrences in cluster I. The value of 100 corresponds to the occurrence of the species in all clusters U_i, whereas the value of 0 demonstrates that the species does not belong to any cluster. **A**: Stability of species occurrences in cluster I. The results of averaging on 100 realizations of clusters U_i (see Section 3.1) are shown. **B**: Reduced stability of species occurrences in cluster I after replacing a part of the marked species by species from other clusters (four marked thermophilic Archaea - *P. horikoshii*, *P. abyssi*, *A. fulgidus*, ans *M. thermoautotrophicum* were replaced by three Archaea from clusters II and III - *A. pernix*, *M. jannaschii*, *S. solfataricus*, and *T. thermophilus*). **C**: Reduced stability of species occurrences in cluster I after reshuffling one of the two sequences which represent each genome (see the text for the procedure of reshuffling).

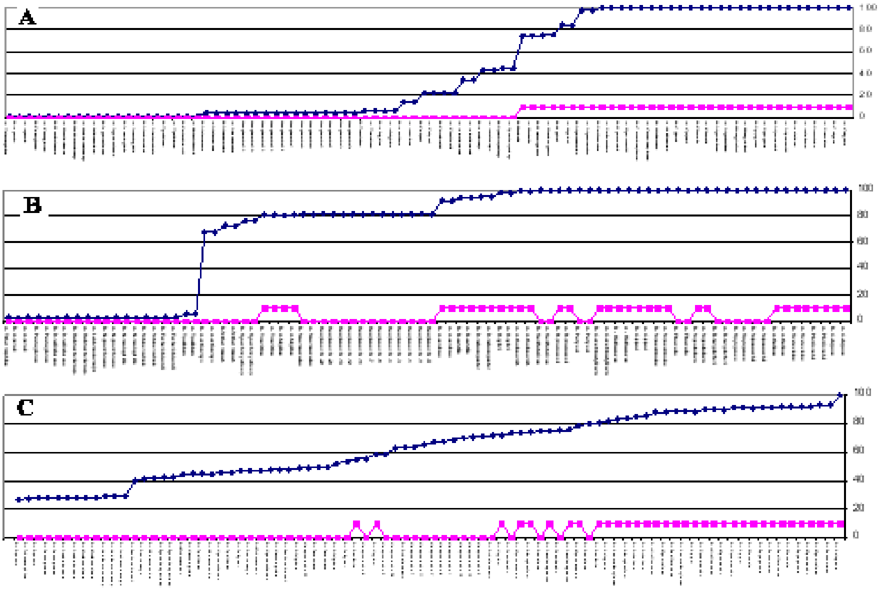


Fig. 6.12 Frequencies of species occurrences in cluster II.**A**: Stability of species occurrences in cluster II. The results of averaging on 100 realizations of clusters U_i (see section 3.1) are shown. **B**: Reduced stability of species occurrences in cluster II after replacing a part of the marked species by species from other clusters (five bacterial genomes - *M. jannaschii*, *S. solfataricus*, *S. pyogenes, B. burgdorferi*, and *E. faecalis* were replaced with five genomes from cluster I - *P. horikoshii*, *A. fulgidus, A. aeolicus, T. maritima*, and *A. thaliana* mit.). **C**: Reduced stability of species occurrences in cluster II after reshuffling one of the two sequences representing each genome(see the text for the procedure of reshuffling).

At this point, it should be noted that there exist, at least, two different genome sequences, A and B, for all of the species in the employed database. For human sequences, 6 out of 12 contigs are considered as A, and 6 as B. In the test procedure, sequence B is not changed, whereas sequence A is transformed in the following way. Let the frequencies of the letters G and C relative to their total number $(G+C)$ in sequence A equal p_G and p_C =1-p_G, respectively. The transformation of sequence A consists in substituting each GC-position (the positions of G or C in sequence A) with the letters G and C independently, the corresponding probabilities being p_G and p_C. Thus, we randomly change the distribution of G and C letters in sequence A without altering the $G+C$ content and the GC positions. The letters A and T in the same sequence A are interchanged in a similar way.

Next, the entire set of sequences A and B is tested in the described above way. The "selected group of species" contains the original elements of cluster I (see Fig. 6.11A). It can be seen in Fig. 6.11C that G↔C and A↔T

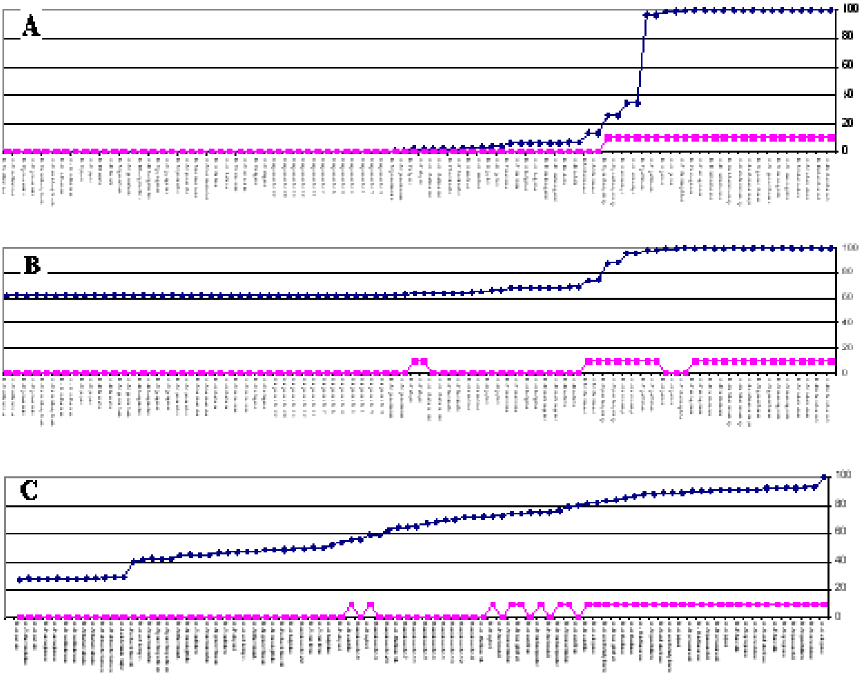


Fig. 6.13 Frequencies of the species occurrences in cluster III. **A**: Stability of species occurrences in cluster III. The results of averaging on 100 realizations of clusters U_i (see section 3.1) are shown. **B**: Reduced stability of species occurrences in cluster III after replacing a part of the marked species by species from other clusters (two bacteria *T.thermophilus*, *A.pernix* were replaced with two bacteria from cluster I and II, *P.abyssi, T.pallidum*). **C**: Reduced stability of species occurrences in cluster III after reshuffling one of the two sequences representing each genome (see the text for the procedure of reshuffling).

permutations totally destroy the previously revealed distinct cluster (compare Figs. 6.11A and 6.11C). The comparison of Figs. 6.12A and 6.13A with Figs. 6.12C and 6.13C, respectively, gives the same results for clusters II and III. To sum up, the tests described above show that the $m-$structure and the corresponding patterns which are formed in $U-$clusters, presumably reflecting some important aspects of the genome similarities, cannot be accounted for by merely the G+C content effects.

6.6 Verification of the Main Pattern in the Case of the (R,Y) Alphabet

In this section, a different procedure of partition verification is described. For the verification of the (R,Y) main pattern, the PAM method (see Appendix

A) is applied for partitioning because this method is more robust and efficient than the well-known k-means algorithm. The CS dataset is partitioned to s clusters where $3 \leq \quad s \quad \leq 9$. For each value of s, the CS is calculated for each of the randomly chosen sets of words $W_i(i{=}1,\ldots,100)$. Then, for each s, the clusters with highly overlapping sets of species across W_i are detected. Further, species with sporadic appearance (with the frequency lower than a certain predefined threshold) are eliminated from the above clusters. The resulting partitions P_s are compared with the main pattern (MP), which was obtained using the WPGMA algorithm. The partition P_s which best corresponds to the MP is selected on the basis of the Cramer correlation coefficient (for the procedure, see [139]). The best correspondence is found for the 8-cluster partition (Table 6.6). In this case, the Cramer correlation coefficient between the $U-$ and PAM-clusters is 0.9537.

Table 6.6 Robust clustering (P_8 partition) of 39 species obtained on the basis of compositional spectra which were calculated using the (R,Y) alphabet.

Cluster 1	%	**Cluster 4**	%
H. sapiens chr. *X*	100	*E. coli*	100
H. sapiens chr. *Y*	97	*D. radiodurans*	99
Mus musculus	100		
A. thaliana	100	**Cluster 5**	
D. melanogaster	65	*N. meningitidis*	100
		N. gonorrhoeae	100
		A. actinomyc.	93
Cluster 2		*H. influenzae*	43
C. elegans	100		
T. thermophilus	100	**Cluster 6**	
P. horikoshii	100	*S. pyogenes*	100
P. abyssi	100	*M. genitalium*	100
A. fulgidus	100	*M. pneumoniae*	100
A. pernix	100	Synechocystis sp	100
B. burgdorferi	100		
H. pylori	100	**Cluster 7**	
M.thermoautotrophicum	100	*B. subtilis*	100
C. jejuni	100	*C. acetobutylicum*	96
A. aeolicus	100	*S. cerevisiae*	96
M. jannaschii	99	*E. faecalis*	71
T. maritima	98	*R. prowazekii*	65
		S. solfataricus	51
Cluster 3			
M. tuberculosis	100	**Cluster 8**	
P. aeruginosa	100	*Leishmania major*	100
Halobacterium sp.	99	*T. pallidum*	75

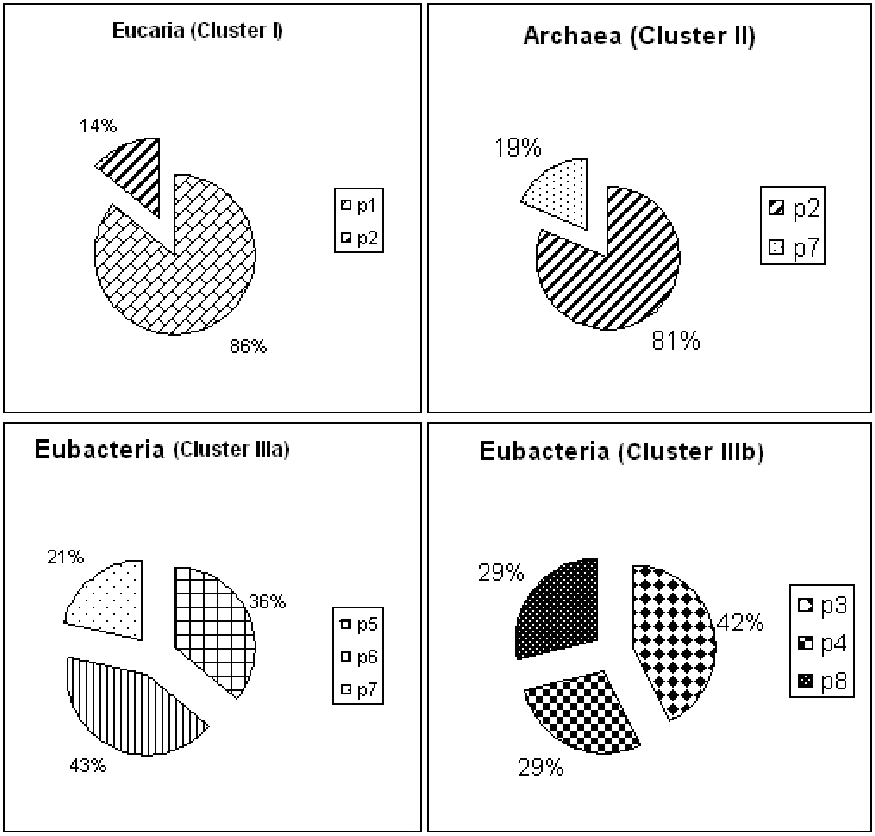


Fig. 6.14 The correspondence between the two clustering schemes of the considered 39 species: Clusters **I-III**: partitions from Fig. 6.9; clusters $\mathbf{p}_1$-$\mathbf{p}_8$ constitute partition P_8(Table 6.6).

We can now construct the MP using the components of P_8 (Fig. 6.14). It can be easily seen that each cluster of P_8 (except for p_2 and p_7) occurs in one of the clusters - I, II, IIIa, or IIIb - of the MP. In other words, with the minor exception of a small part of p_2 and p_7, the main pattern can be viewed as being constructed of elementary P_8 blocks. This consistency verifies the actual existence of the MP structure.

6.7 The List of Depicted Genomic Sequences

The notations of genome fragments in all the above figures corresponds to the indexes of the genome sequences listed below. Every record in the list consists of the species name and some data about the fragment (in brackets); an accession number for some fragments is shown to avoid wrong identification. For a fragment of a complete genome, the starting point and the fragment length are specified; for a fragment having an accession number, only its length is specified. The value between 0 and 1 relates to the $G + C$ content of the fragment. For example, the record "*Mycobacterium tuberculosis* (A: 1199341 (358717), 0.65; B: 2579341 (358913), 0.65)" means that *M. tuberculosis* is presented by two fragments, A and B, from the complete genome; the fragment A starts at position 1199341 and has the length of 358717 bp, while the fragment B starts at position 2579341 and has the length of 358913 bp; both fragments have the $G + C$ content equal to 65%.

Eukaryota

Homo sapiens chr. X (NT 011528) (539188, 0.40); *Homo sapiens* chr. Y (NT011864) (539595, 0.40); *Homo sapiens* chr. 1 (NT 004302) (539495, 0.36); *Homo sapiens* chr. 3 (NT 002444) (543554,0.46); *Homo sapiens* chr. 4 (NT 006051) (535847, 0.44); *Homo sapiens* chr. 6 (NT 007122) (599072, 0.43); *Homo sapiens* chr. 7 (NT 007643) (447791, 0.36); *Homo sapiens* chr. 11 (NT 008933) (506578 , 0.37); *Homo sapiens* chr. 13 (NT 009796) (537882, 0.38); *Homo sapiens* chr. 20 (NT 011328) (540270, 0.44); *Homo sapiens* chr. 22 (NT 001454) (701877, 0.50); *Mus musculus* (A: chr7, AC012382, 276523, 0.44; B: chr.11, AL603707, 234182, 0.49); *Caenorhabditis elegans* (A: chr1, 1-438825, 0.36; B: chr2, 1-350000 , 0.36); *Drosophila melanogaster* (A: chr.2 AE003641, 299556, 0.42; B: chr. X, AE003506, 300000, 0.43); *Arabidopsis thaliana* (A: chr.1, NC 003071.1 100000-531319, 0.36; B: chr.1, NC 003075.1, 400000-727859, 0.36); *A. thaliana* mitochondrial genome (A: NC 001284.1, 366923, 0.45); *Saccharomyces cerevisiae* (A: chrii, 1-800000, 0.38; B: chrxv,1-800000, 0.39); *Leishmania major* (A: AE001274, 1-171770, 0.62). **Eubacteria** *Bacillus subtilis* (A: 1199941 (579647), 0.43; B: 2219941 (399002), 0.41); *Streptococcus pyogenes* (A: 239941 (690238), 0.38; B: 1079941 (696345), 0.39); *Mycoplasma genitalium* (A: 1 (287593), 0.33; B: 278581 (288000), 0.30); *Mycoplasma pneumoniae* (A: 239941 (199523), 0.40; B: 539941 (199523), 0.40); *Mycobacterium tuberculosis* (A: 1199341 (358717), 0.65; B: 2579341 (358913), 0.65); *Synechocystis sp* (A: 719941 (349960), 0.48; B: 2699941 (350000), 0.47); *Helicobacter pylori* (A: 599941 (320335), 0.39; B: 1439941 (320387), 0.39); *Escherichia coli* (A:599941 (519942), 0.51; B: 2999941 (542976), 0.51); *Deinococcus radiodurans*(A: 599941 (399971), 0.67; B: 1799941 (399983), 0.66); *Thermotoga maritima* (A:59941 (370054), 0.46; B:1259941 (366191), 0.46); *Aquifex aeolicus* (A: 599941 (399976), 0.43; B: 1199941 (400002 , 0.44); *Neisseria meningitides* (A: 599941 (361259), 0.51; B: 1199941 (373905), 0.52); *Neisseria gonorrhoeae* (A: 350020, 0.53 ; B : 355192, 0.54) ; *Campylobacter jejuni* (A: 59341 (399984), 0.31; B: 1079341 (400002), 0.30); *Haemophilus influenzae* (A: 119941 (399863), 0.38; B: 1139941 (399981), 0.38); *Clostridium acetobutylicum* (A: 315781 (347567), 0.32; B: 3000001 (340105), 0.31); *Treponema pallidum* (A: 59941 (275933), 0.52; B: 719941 (275984), 0.53); *Pseudomonas aeruginosa* (A: 720001 (345206), 0.65; B: 1920001 (355163), 0.67); *Actinobacillus actinomycetemcomitans* Strain HK1651 (A: 6000 (338681), 0.45; B: 800000 (344947), 0.45); *Rickettsia prowazekii* (A: 239941 (276000), 0.29; B:719941 (276000), 0.29); *Borrelia burgdorferi* (A: (1) 399967, 0.29; B: 400000 (399979), 0.29). **Archaea** *Halobacterium sp* .NRC-1 (A: 240001 (191652), 0.62; B: 64001 (211652), 0.64); *Pyrococcus horikoshii* (A: 240001 (399992), 0.42; B: 840001 (400002), 0.42); *Pyrococcus abyssi* (A: 360000 (360000), 0.45; B: 1200001 (360000), 0.45); *Archaeoglobus fulgidus* (A: 12061 (399986), 0.48; B: 1200061 (400002), 0.48); *Methanococcus jannaschii* (A: 120001 (399868), 0.32; B: 840001 (399977), 0.31); *Methanobacterium thermoautotrophicum*(A:599941 (344374), 0.49; B: 1199941 (344455), 0.50); *Aeropyrum pernix* (A: 360061 (400002), 0.58; B: 960061 (400002), 0.57); *Sulfolobus solfataricus* AE006641 (A: 400001 (584947), 0.36; B: 1220002 (308779), 0.36)

Chapter 7
Marker-Function Profile-Based Clustering

7.1 General Description of the Profile-Construction Method

In Chapter 5, it was shown that any text can be viewed as a stream of overlapping N-grams. We have seen that N-gram techniques of sequence analysis are based on the reduction of the whole text to the vector of length L^N, where L is the size of the alphabet and N is a predefined length of the N-grams. The coordinates of such vectors can be obtained from a certain formula using either the observed N-gram frequencies or the ratios of observed to expected frequencies. In what follows, we describe a different procedure - the conversion of a text $\mathfrak{T}$ to a point in the K-dimensional Euclidean space for the purpose of further clustering.

- Consider a text $\mathfrak{T}$ of size $\mathfrak{T}_N = |\mathfrak{T}|$.
- M specified positions (markers) in the text $\mathfrak{T}$ constitute a set $mrk(\mathfrak{T}) = (m_1, m_2, \cdots m_i, \cdots m_M)$, where i is the index of markers m in the set of markers mrk for $\forall i, 1 \leq i \leq M, 1 \leq m_i \leq \mathfrak{T}_N$. It is assumed that the marker positions are non-random and that the sequence fragments in the vicinity of the markers have some mutual lines of similarity.
- Two profile parameters, a and b, are introduced to set the margins of a marker neighbourhood. Although the similarity between the marker neighbourhood fragments is difficult to observe, it is assumed that there exists a function f, defined on the sequence strings, which reflects the distribution of a certain numerical feature of the substrings in the marker neighbourhood.
- Let the function f be such that for $\forall s$, $f(s) \rightarrow R$, which means that for any string s $f(s)$ is defined and its value is a real number. It is our intention to introduce an approach based on the characteristic distribution of the function f around an averaged marker $m_i, m_i \in mrk$. Theoretically, any function f defined on all strings s over an alphabet $\mathfrak{A}$ can be used for further mapping.

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 113–145, 2010.
springerlink.com

- The parameter w which represents the size of a sliding window is introduced.
- Mapping $F(f, \mathfrak{T}, i, w)$ is defined in such a way that the value of F is equal to that of the function f, the argument being the string s of size w. F is the mapping of the string $\mathfrak{T}$ onto a numerical sequence X. In what follows, $substr(s, i, k)$ denotes the well-known function that returns the substring of length k which starts from position i within s. Let us introduce function $\Psi(i) = F(f, \mathfrak{T}, i, w) = f(substr(\mathfrak{T}, i - [w/2], w))$, which is defined over the segment $[[w/2] + 1, N - [w/2]]$. The profile $p(f, \mathfrak{T}, mrk(\mathfrak{T}), w)$ is defined over the segment $[-a, b]$ as

$$p(x) = \frac{\sum_{j=1}^{m} \Psi(m_j + x)}{M} = \frac{\sum_{j=1}^{M} f(substr(\mathfrak{T}, m_j - [w/2], w)}{M}. \tag{7.1}$$

  In this chapter, two examples of such profiles are presented: in the first example, the function $f(s)$ is chosen as the Linguistic Complexity measure [261], in the second example, the measure of DNA curvature [234] is employed.
- After introducing the distance measure

$$d(p_1(f, \mathfrak{T}_1, mrk(\mathfrak{T}_1), w), p_2(f, \mathfrak{T}_2, mrk(\mathfrak{T}_2), w))$$

  between two profiles p_1 and p_2, one can perform clustering of the set of texts $\mathfrak{T}_1, \mathfrak{T}_2, \mathfrak{T}_k, \mathfrak{T}_L$ using any clustering method.

7.2 Eukaryotic Genome Tree Based on Linguistic Complexity Profiles

7.2.1 Sequence Complexity Measures

Different methods based on sequence (compositional) complexity measurements, which may be used for overall characterization and comparison of long genomic sequences (see [21] and [149], [148] for review), have been proposed since the early stages of bioinformatics [34], [150], [220], [221], [222], [235], [258], [261]. These methods can be applied even in the case of low similarity of sequences [147]. It was shown [149],[148] that the numerical value of compositional complexity of a sequence depends solely on the alphabet size and on the frequencies of the occurrences of certain elements (monomers, dimmers, or N-grams over the chosen alphabet). The degree of sequence interrelation within the set is of major importance. For example, when a lot of sequences in the set have a large number of short subsequences in common, the question arises as to whether these sequences are functionally or evolutionary related. The significance of the pertinent research results can be assessed by comparing them with those obtained for randomly generated data. There also exists an alternative approach to the study of the compositional complexity of a sequence. According to this approach, the average behavior of a number of distinct substrings in the text is studied, which allows introducing a special measure, called the *complexity index.* This measure reflects the richness of the language of the sequence. For example, sequences with relatively low complexity indexes contain a large number of repeated substrings. To identify text fragments of unusually low or high complexity, one should determine the deviations of the complexities of the text fragments from the average or maximum sequence complexity. These position-dependent complexity deviations can help characterize genomic sequences. In Section 7.1, the construction of the profile $p(f, \mathfrak{T}, mrk(\mathfrak{T}), w)$ was described. Genetic sequences may be characterized using the function f based on the sequence complexity measure as it will be shown below.

7.2.2 Construction of Genomic Linguistic Complexity Profiles

To describe the procedure of profile construction (see above and Equation 7.1), the following parameters should be introduced: (i) text $\mathfrak{T}$; (ii) a set of M specified positions $\{m_i\}$; (iii) the size of the sliding window w; (iv) the interval $[-a, b]$; and (v) function f. Below, we describe these parameters, which are further used for clustering of chromosomes of several eukaryotic genomes. Text $\mathfrak{T}$ represents the sequence of one complete eukaryotic chromosome from the GenBank, e.g., chromosome 4 of *Drosophila melanogaster*. In Table 7.1, we list all the chromosomes used for the linguistic complexity (LC) study. The chromosome sequences of these 11 eukaryotic organisms are taken from the site of the National Center of Biotechnology Information ($ftp://ftp.ncbi.nih.gov/genbank/genomes$). As the set of markers we choose the set of all start codon positions. These starts are also obtained using the annotations of complete eukaryotic chromosomes in the GenBank. In what follows, the average LC profiles around the starts of coding sequences (CDS) will be called, for short, *LC profiles.* The terminology used below, which is related to the gene structure, was introduced in Chapter 1.

The size of the sliding window w is set to be 50 bp. This value was empirically chosen from the tested sizes of 20, 50, 100, and 200 bp since it appears to highlight in the best way the most common LC features for all the individual genomes. The interval $[-a, b]$ is chosen to be $[-200, +200]$. Generally speaking, our goal is to use the LC variation around the start of translation. The size of the translation start neighbourhood was determined empirically to be $[-200 : +200]$ for every gene. It appeared that the choice of the parameters a and b had a minor effect on the results. It was much more important to strictly distinguish between coding and non-coding regions (see Fig. 1.7). Thus, a and b are equal to 200 only in the cases where all 200 downstream nucleotides are located in a coding region, while all 200 upstream nucleotides are located in a non-coding region. Otherwise, only the relevant downstream coding or

Table 7.1 List of the employed species and the processed chromosomes.

N	Name of organism	Kingdom	Number of chromosomes	Processed chromosomes
1	*Arabidopsis thaliana*	Plant	5	1-5
2	*Candida glabrata*	Fungi	13	1-13
3	*Cryptococcus neoformans*	Fungi	14	1-14
4	*Drosophila melanogaster*	Animal	5+X	2L, 2R, 3L, 3R
5	*Homo sapiens*	Animal	22+XY	1-10
6	*Mus musculus*	Animal	19+XY	1-10
7	*Oryza sativa*	Plant	12	1-12
8	*Ostreococcus lucimarinus*	Plant	10	1-10
9	*Plasmodium falciparum*	Animal	10	1-14
10	*Saccharomyces cerevisiae*	Fungi	16	1-16

upstream non-coding nucleotides are used. The start profile is constructed using only the genes with the first coding exon longer than 25 nucleotides (1/2 of the window size) flanked by an upstream non-coding region (see Fig. 1.7) longer than 25 nucleotides. For example, if the length of the upstream region is 723 bp and the length of the first coding exon is 115 bp, the LC profile for such gene is obtained starting from position -200 and up to position +90. The flanked windows that overlap with the neighbouring region are not considered in further calculations in order to prevent the impact of boundary regions. Function $f(s)$ is chosen as the *Linguistic Complexity measure* [261]. It is defined as the ratio of the actual number of different substrings present in the string s to the maximum possible number of substrings in the string s over the same alphabet. The maximum number of different substrings is calculated according to the following formula:

$$\sum_{k=1}^{m} min(4^k, m-k+1), \tag{7.2}$$

where 4 is the number of letters in the DNA alphabet; $m = |s|$. The algorithm used in (Troyanskaya et al., 2000) provides an effective way of calculating LC profiles. Since the window size is 50 bp, the value of k from formula 7.2 is equal to 50, which gives

$$\sum_{k=1}^{50} min(4^k, 50-k+1) = 4 + 16 + \sum_{k=3}^{5} 0(51-k) = 20 + 24 \times 47 = 1148.$$

The mean LC profiles of chromosomes are obtained by averaging the profiles of all relevant genes from the same individual chromosome. The standard errors are estimated by the bootstrap method using 1,000 runs.

7.2.3 Peculiarities of Different Eukaryotic Genomes Derived from Their LC Profiles

LC Profiles of *Drosophila melanogaster* and *Caenorhabditis elegans* genomes. Fig.7.1 shows the LC profiles of the chromosomes $2L$, $2R$, $3L$, and $3R$ of *D. melanogaster* and of the chromosomes 1-5 of *C. elegans*. It is easy to see that the LC profiles related to different chromosomes of the same organism are very similar to each other. On the other hand, there are certain features of LC profiles that are common to practically all eukaryotic profiles studied by such techniques (see below). For example, a) the average LC of upstream intergenic regions is typically lower than that of coding regions; b) LC in the close vicinity of the translation start is lower due to the increased number of regulatory signals in this area. Consequently, a local minimum is observed close to zero (i.e., to the start of translation); c) the global maximum of the function is typically located between zero and +25 bp.

LC and GC composition of Upstream Non-translated Regions. It was noted [261] that LC profiles exhibit significant shape peculiarity around translation starts for GC-rich[1] and AT genomes. The latter are genomes that have the AT content of CDS higher than 48%. AT genomes demonstrate the similar main feature of lower LC values for non-coding regions as compared to coding regions. Similar to the case of AT prokaryotic genomes, eukaryotic coding sequences have the AT content higher than 50%. Therefore, upstream non-coding sequences are AT-richer, which automatically lowers the LC values. By definition, sequences with the highest LC values have the most balanced oligonucleotide composition. Non-coding sequences are less balanced even at the level of mononucleotides. Thus, LC decreases with the decrease of GC content. Of course, the trends of LC variation only follow the trends of GC content variation, but are not necessarily identical with them. For *D. melanogaster*, the average LC values at position -100 in the intergenic regions and at position +100 in the coding sequences are 0.969±0.0004 and 0.976±0.0003, respectively (Table 7.2). The difference between these two regions for *D. melanogaster* is smaller than those for *A. thaliana* and *C. elegans*, but still pretty significant.

Start Signals of Translation. First of all, we would like to compare LC profiles constructed for prokaryotic genomes with those that characterize eukaryotic chromosomes. In many prokaryotic AT genomes there are local complexity minima immediately before the start of CDS [261]. However, in some cases (e.g., *E. coli*, *H. influenzae*, and *M. pneumoniae*), these minima are not clearly manifested [261]. From the data presented in Figs. 7.1 and 7.2, it can also be seen that, in contrast to the case of prokaryotic AT genomes, eukaryotic LC profiles do not show significant decline before the start of CDS. The only exception is the LC profile of *A. thaliana*, which is slightly similar

[1] Genomes, in which the total content of G and C in the genomic sequence is more than 50% are called GC-rich genomes.

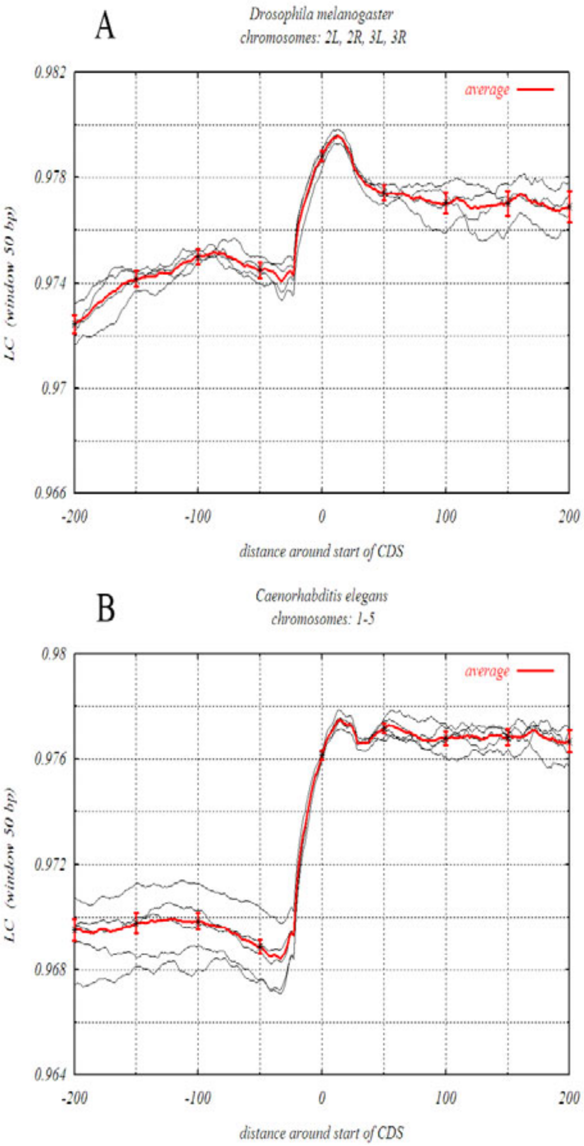


Fig. 7.1 LC profiles in the neighbourhood of the start of the coding sequences for **A**: *D. melanogaster* and **B**: *C. elegans* chromosomes. Black lines represent average LC profiles of individual chromosomes. The bold red solid line represents the LC profile averaged over all the chromosomes of the same organism. Standard deviations are estimated by the bootstrap method using 1,000 runs.

Table 7.2 Average linguistic complexity in coding and non-coding regions of particular species.

N	Name of Organism	Average LC value at position -100 in non-coding regions	Average LC value at position +100 in coding regions	Local minimum LC value and its position in the vicinity of the CDS starts	Maximum LC value and its position in the vicinity of the CDS starts
1	*Arabidopsis thaliana*	0.969±0.0004	0.976±0.0003	0.966 / -34	0.976 / +18
2	*Caenorhabditis elegans*	0.970± 0.0003	0.977±0.0002	0.968 / -34	0.977 / +15
3	*Candida glabrata*	0.974±0.0008	0.978±0.0007	0.972 / -23	0.979 / +16
4	*Cryptococcus neoformans*	0.976±0.001	0.977±0.0009	0.974 / -41	0.979 / +9
5	*Drosophila melanogaster*	0.975± 0.0003	0.977±0.0004	0.974 / -33	0.980 / +13
6	*Homo sapiens*	0.969±0.0008	0.973±0.0009	0.969 / -46	0.975 / +10
7	*Mus musculus*	0.970±0.0009	0.973±0.001	0.970 / -77	0.976 / +7
8	*Oryza sativa*	0.965±0.0009	0.964±0.0007	0.958 / -25	0.966 / +10
9	*Ostreococcus lucimarinus*	0.944±0.002	0.960±0.002	0.944 / -50	0.959 / +16
10	*Plasmodium falciparum*	0.882±0.006	0.961±0.001	0.870 / -69	0.964 / +24
11	*Saccharomyces cerevisiae*	0.972±0.001	0.978±0.0007	0.973 / -23	0.979 / +24

to the prokaryotic profiles in that there is a definite decrease of LC values from 0.969±0.0003 at position -100 to 0.966±0.0003 at positions -30 ±5.

Linguistic Complexity at the Starts of Translation. All four LC profiles of *D. melanogaster* chromosomes (Fig. 7.1 A) have a global maximum at position +20 (which is also observed in the *A. thaliana* LC profiles). This maximum can be accounted for by the difference in dinucleotide frequencies for the coding and non-coding chromosome regions [31].

Average LC profiles. In Fig. 7.1, the LC profiles of all the chromosomes of two organisms (*D. melanogaster* and *C. elegans*) are presented, while the average LC profiles of 11 genomes are shown in Fig. 7.2. Such species-consensus LC profiles reflect important LC profile characteristic features of all chromosomes of the same genome. The species are divided into three separate groups (A, B, C) according to the similarity of their LC profile shapes and ranges. First, we will describe the *P. falciparum* profile (Fig. 7.2A), which is essentially different from the profiles of other 10 species.

LC Profile of *Plasmodium falciparum.* Comparing the data, presented Fig. 7.2A and in Figs. 7.2B,C, the difference in the Y-axis scale should be noted (the range being [0.84:1.0] and [0.964:0.980], respectively). The extremely low sequence complexity values for intergenic regions of *P. falciparum* can be accounted for by a very high AT content of these regions [82], [204]. The LC values in the non-coding region lie below the level

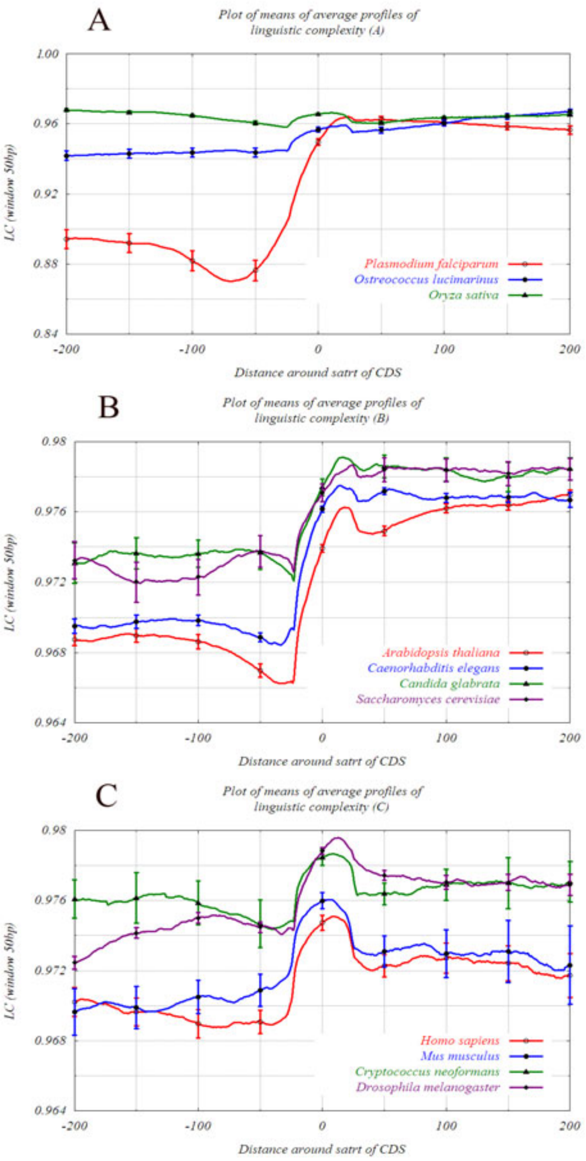


Fig. 7.2 Average LC profiles of 11 eukaryotic organisms (see Table 7.1). **A**: *P. falciparum*, *O. lucimarinus*, and *O. sativa*; **B**: *A. thaliana*, *C. elegans*, *C. glabrata*, and *S. cerevisiae*; **C**: *H. sapiens*, *M. musculus*, *C. neoformans*, and *D. melanogaster*. Standard deviations are estimated by the bootstrap method using 1,000 runs.

of 0.9, being much lower than the corresponding values for other organisms presented here. The LC values in the coding region are pretty similar to those observed for *O. sativa* and *O. lucimarinus*. The difference between the LC average values in the coding and the non-coding region is one-fold greater than that for other genomes. The position of the minimum in intergenic region is located at 70 bp upstream from the translation start, which is also different from the minimum positions for other genomes.

LC profiles of *Oryza sativa* and *Ostreococcus lucimarinus*. The average LC profile of *O. sativa* is very different from those of all other species shown in Fig. 7.2. First, there is practically no difference between the mean LC values of the intergenic and the first exon regions; second, the profile exhibits strong decline along the upstream region and strong rise along the first exon region (see Table 7.1: 0.965±0.0009 at -100 position vs. 0.964±0.0007 at +100 position). These interesting features certainly require further investigation. The LC values vary from 0.968 to 0.958 in the intergenic regions and are about 0.964 along the first exons. The average LC profile of *O. lucimarinus* also differs from those of all the other 10 species under consideration. Contrary to the LC profiles of *O. sativa* and *A. thaliana*, it has no minimum in the intergenic region. The average LC values of *O. lucimarinus* vary around 0.942 in the upstream region. In addition, the profile shows a significant continuous rise (from 0.955 to 0.965) along the coding sequence (the first exons). The local maximum at the border of the coding and the non-coding regions is observed.

Comparison of *Arabidopsis thaliana* and *Caenorhabditis elegans* LC profiles. The LC profiles of two well-annotated genomes, those of *A. thaliana* and *C. elegans* appear in Fig. 7.2B. Although the two species belong to different eukaryotic groups, their average LC profiles have some significantly similar features. For both genomes, the LC values are in the range of 0.969÷0.970 and 0.976÷0.977 in the 5′-noncoding and in the coding regions, respectively. The shapes of the LC profiles in the neighbourhoods of the translation starts also look very similar; in particular, their minima are located at the same position (-30 bp).

LC Profiles of *Saccharomyces cerevisiae*, *Candida glabrata*, and *Cryptococcus neoformans* fungi species *S. cerevisiae* and *C. glabrata* (Fig. 7.2B), which are related organisms, have similar LC profiles. In particular, they have close LC values (0.974 and 0.978, respectively) in the intergenic region along the first exons. Both profiles exhibit a minor decline at the -25 bp position and show no significant maxima at the border of the intergenic and the coding regions. The *C. neoformans* LC profile differs from those of the two other fungi genomes. It is more similar to the profiles of *D. melanogaster*, *H. sapiens*, and *M. musculus* and, therefore, appears in Fig. 7.2C. The *C. neoformans* LC values are 0.976 in intergenic region and 0.977 along the upstream sequences. The LC profile has a minimum at the -40 bp position and a dominating maximum at the +1 bp position. Unlike the average LC values, the shape of the *C. neoformans* profile is similar to those of the of *H. sapiens* and *M. musculus*.

LC profiles of *Homo sapiens* and *Mus musculus.* The LC chromosome profiles and the averaged *Homo sapiens* and *Mus musculus* genome profiles are very similar to each other. Their LC values are practically identical in the regions upstream and downstream from the translation starts: 0.970 in the intergenic region and 0.973 in the coding sequences (Fig. 7.2C and Table 7.1). The differences between the LC values in the intergenic and coding sequences are smaller than those for the profiles presented in Fig. 7.2B. No minimum is observed before the start of translation.

7.2.4 LC Dendrograms

A dendrogram is a tree diagram used to illustrate the nested structure of the clusters produced by a clustering algorithm. Dendrograms are often used in computational biology to illustrate clustering of genes. Here the clustering of LC profiles is illustrated by means of dendrograms (see Chapter 2 for the details of the dendrogram approach). The Euclidian distance $d_{1,2}$ for each pair of LC profiles is calculated as

$$d_{1,2} = \sqrt{\frac{\sum_{i=1}^{L}(x_{1,i} - x_{2,i})^2}{L}}, \tag{7.3}$$

where $x_{1,i}$ and $x_{2,i}$ are the LC values in the i-th position of profiles 1 and 2, respectively; L is the length of profiles, equal to 401. The Euclidean distance or the squared Euclidean distance are used as the dissimilarity measures. For data clustering, Unweighted Pair Group Arithmetic Average Clustering (UPGMA) and the k-means method are used (for details, see Chapter A). UPGMA, which is also called Average-Linkage Clustering, minimizes the objective function equal to the sum of squares of the distances to determine the clustering that results in the lowest sum. Thus, UPGMA is an agglomeration method, which employs a sequential clustering algorithm, identifying local topological relationships in the order of similarity. UPGMA is applied to the construction of dendrograms using the Software PHYLIP package [73].

The results of clustering are presented in Fig. 7.3 in the form of dendrograms. It appears that the Euclidian distances between LC profiles of different organisms are of different order of magnitude (see Table 7.4). It could be expected, in particular, from the comparison of the LC profiles of *Homo sapiens* and *Mus musculus* chromosomes (see above) that LC profiles, corresponding to chromosomes of the same species, should be similar to each other, thus forming an individual cluster. It could be also expected that chromosome LC profiles for different species should be dissimilar and belong to different clusters. Indeed, the distances between LC profiles of two distantly related species are one or two orders of magnitude bigger than the distances between

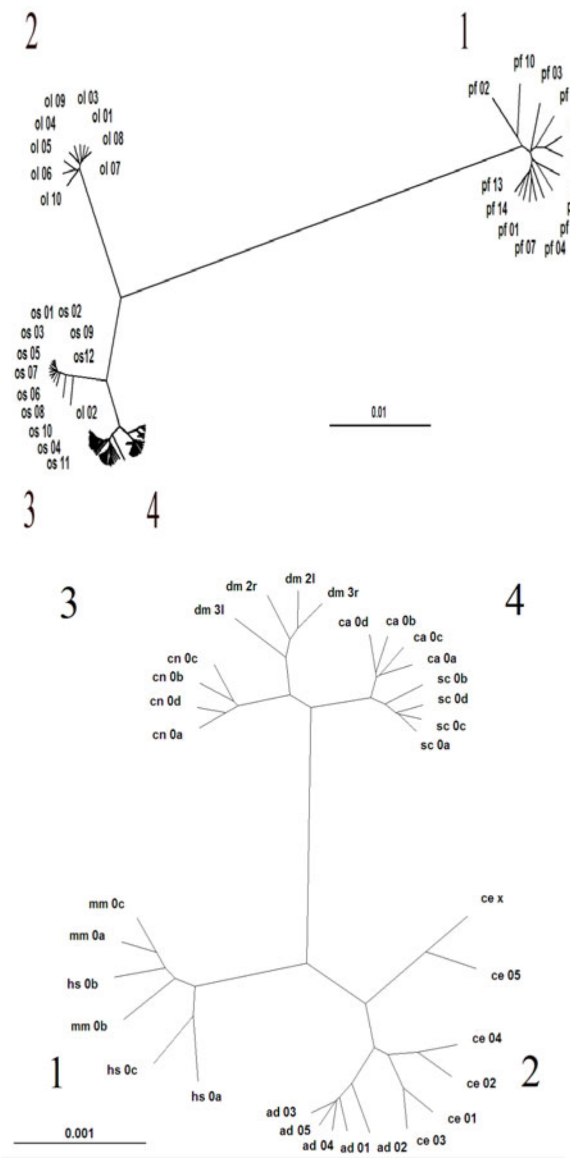


Fig. 7.3 LC dendrograms. The abbreviations used: *Arabidopsis thaliana* = ad; *Caenorhabditis elegans* = ce; *Candida glabrata* = ca; *Cryptococcus neoformans* =cn; *Drosophila melanogaster* = dm; *Homo sapiens* = hs; *Mus musculus* = mm; *Oryza sativa* = os; *Ostreococcus lucimarinus* = ol; *Plasmodium falciparum* = pf; *Saccharomyces cerevisiae* = sc.

Table 7.3 Construction of the four types of common profiles by combining profiles of different organism.

Organism	Included profiles (chromosomes)	Label of common profiles
Candida glabrata	1-4	a
	5-7	b
	8-10	c
	11-13	d
Cryptococcus neoformans	1-3	a
	4-6	b
	7-10	c
	11-14	d
Homo sapiens	1-3	a
	4-6	b
	7-10	c
Mus musculus	1-3	a
	4-6	b
	7-10	c
Saccharomyces cerevisiae	1-4	a
	5-8	b
	9-12	c
	13-16	d

LC profiles of the same organism. However, the distances between LC profiles of closely related species can be comparable to or even smaller than the interspecies distances. Four distinct groups are observed in the dendrogram based on 113 profiles of 11 eukaryotic organisms (Fig. 7.3A), three groups corresponding to individual organisms: *P. falciparum*, *O. lucimarinus*, and *O. sativa.* A detailed dendrogram for the fourth group of other eight organisms is shown in Fig. 7.3B, where four subgroups are clearly observed. One can see that the composition of two subclusters appears to be quite natural. The related organisms belong to the same subgroups: *H. sapiens* and *M. musculus* chromosomes form cluster 1, while *C. glabrata* and *S. cerevisiae* form cluster 4. Other two groups comprise taxonomically distant organisms, even belonging to different kingdoms: *A. thaliana* and *C. elegans* (plant and animal) form cluster 2; and *C. neoformans*, while *D. melanogaster* (fungi and animal) form cluster 3.

Table 7.4 Euclidian distances between four general clusters of 113 profiles.

No	No. 1	No. 2	No. 3	No. 4
No. 1	0.000000	0.038313	0.052859	0.059505
No. 2	0.038313	0.000000	0.015412	0.023597
No. 3	0.052859	0.015412	0.000000	0.010949
No. 4	0.059505	0.023597	0.010949	0.000000

7.2.5 Comparison between LC Dendrograms and Taxonomic Cladograms

Some of the above results are organized in Fig. 7.4A, B in the form that emphasizes the similarities and the differences between the obtained partitions (Fig. 7.3) and the classical eukaryotic taxonomy. There are several differences in the topology of the dendrograms *A* and *B*; for example, in contrast to the dendrogram corresponding to the natural biological taxonomy (Fig. 7.4A), the three considered plants, *A. thaliana* (ad), *O. lucimarinus* (ol), and *O. sativa* (os), do not form a monophyletic group in the LC dendrogram (Fig. 7.4.B). It should be noted that the LC profile of the *O. lucimarinus* alga is very different from all other profiles. *Ostreococcus* is one of the smallest known eukaryotic organisms, about 1 micron in size; it is a unicellular organism belonging to an early-diverging class within the green plant lineage. The most striking feature of *O. lucimarinus* and the related species is their minimal cellular organization. It can be suggested that the properties of the *O. lucimarinus* LC profile may be accounted for by the morphological and taxonomic peculiarities of this species.

From the data presented in Fig. 7.2A and Fig. 7.3A it can be seen that the LC profile of the rice genome (os) is quite different from all the other profiles though it could be expected to be very similar to the *A. thaliana*

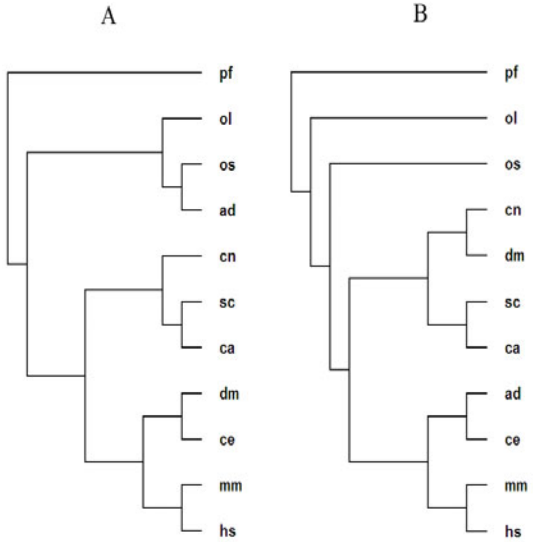


Fig. 7.4 LC dendrogram and phylogeny of 11 eukaryotic organisms listed in Table 7.1.

A: The dendrogram corresponds to the widely accepted taxonomy of the selected eukaryotes (NCBI taxonomy page http://www.ncbi.nlm.nih.gov/Taxonomy/).

B: The dendrogram corresponds to those presented in Fig. 7.3.

profile. Among the possible reasons of such dissimilarity may be just incorrect annotation. In any case, from the results of clustering presented in Fig. 7.4, it can be concluded that further research should be done to establish the reasons of differences between the natural taxonomy and the classification based on the LC profiles of the plants *A. thaliana* (ad), *O. lucimarinus* (ol), and *O. sativa* (os). In Fig. 7.4B, the plant *A. thaliana* (ad) is paired with the worm *C. elegans* (ce), while the fly *D. melanogaster* is paired with the encapsulated yeast-like fungus *C. neoformans* (cn). Future research should answer the question as to whether there exists a biological explanation of such partition.

Classification based on the Euclidean distances between LC profiles seems to correlate with conventional taxonomy. We speculate that the sequence structure manifested through average linguistic complexity profiles is closely related to the evolution of various species and their genomes.

7.2.6 Discussion of the Results

In almost all LC profiles one can observe a sunken area in a pre-translation region. The most likely explanation for the decrease in average LC values is the presence of repetitive sequences that are utilized in the initiation of translation. The locations of these sunken areas point to species-specific upstream regions that are especially "loaded" with repetitive regulatory sites related to the regulation of translation. An alternative but less likely explanation connects the repetitiveness of sequences with the regulation of transcription.

Most LC profiles have convex shapes around the start of translation (TS region). For example, such a shape is very evident for the profiles presented in Fig. 7.2C: for the profiles of *H. sapiens* and *M. musculus*, and for the profiles of *C. neoformans* and *D. melanogaster*. "Pure" intergenic and "pure" genic LC values are pretty comparable along the profiles of these species, demonstrating a similarity in richness of genic and intergenic vocabularies; mixed border regions of coding and intergenic sequences are characterized by a rise (a convex shape) in LC values. One possible explanation is that by merging two different vocabularies together when a running window overlaps translation starts brings substantial enrichment of the repertoire of N-grams into use. The genomes presented in Fig. 7.2B have different TS convex shapes from those presented in Fig. 7.2C. The main characteristic of the profiles presented in Fig. 72B is in the dramatic elevation of the curves from the lower level (typical to intergenic regions) to a significantly greater complexity (typical to first exons).

Interestingly, behaviors of LC profiles along the coding regions are different for different genomes: there is complexity elevation (positive correlation of LC with the distance from translation starts) for *A. thaliana, O. lucimarinus,* and *O. sativa*; there is complexity decline (negative correlation) for *P. falciparum*; and the most common behavior is expressed in insignificant fluctuations around a mean LC value typical for genic sequences of a certain genome. Unfortunately, we cannot provide any biological explanation for these genomic peculiarities at the moment.

It was expected that LC profiles corresponding to chromosomes of the same species should be similar and form individual clusters, while chromosomal LC profiles of varied species should be dissimilar and belong to different clusters. This expectation was fulfilled only in part: the LC profile technique is not able to separate chromosomes belonging to different mammals.

The observations reported here suggest that the average linguistic complexity analysis of putative promoter structures can yield new insight into the nature of the genome. The data reported here indicate that entire genomic sequences can be analyzed in efforts to gain an understanding of the evolutionary relationship between various species and among chromosomes within a single species. As described here, the average linguistic complexity profile of genomic structure reveals a great deal of fine structure variations in the various sequences available.

7.2.7 Conclusions

The results show both general and genome-specific linguistic features of the promoter organization of eukaryotic genes. The most general feature found in all genomes is lower average complexity of the regions upstream of the translation start compared to the average complexity of the first exons. In addition, in almost all of the genomes studied in this investigation, LC profiles have convex shapes around the start of translation. While these statistical laws of LC profile behaviors in promoter regions are quite general, clustering based on the Euclidean distances between LC profiles, as a rule, results in collecting chromosomes of the same species in an individual cluster; chromosomal LC profiles of varied species appeared to be dissimilar as they are located in different clusters.

Clearly, promoter structures of eukaryotic genes have many important parameters that were not considered in this Chapter; they are left for future research.

7.3 Prokaryotic Species Classification Based on DNA Curvature Distribution

In the previous section an example of classification of eukaryotic chromosomes was introduced. Here, we present another example of the profile-construction method. A description for a specific procedure of profile construction of a few parameters should be introduced, including the function f on which the whole procedure is based. In this subsection an estimation of the DNA curvature is chosen as such a function.

7.3.1 DNA Curvature Prediction

Among different local DNA characteristics there is a so-called "curved DNA" property. A simple meaning of the term is that in a certain region of the DNA molecule a curved (or a curvilinear) line describes the DNA helical axis better than a straight line. There are a few different approaches to predict the helical axis of the DNA molecule. Accordingly, there are a few software products intended to predict a DNA path, which is, actually, a prediction of the whole sequence-dependent static DNA structure. One of the most widely used programs is CURVATURE [260, 256, 22, 234]. An algorithm is based on the "nearest neighbor model" or the "wedge model" [260, 256, 22, 234] and is currently, broadly used [134, 11, 107, 276, 92]. The software is applicable in plotting the sequence-dependent spatial trajectory of the DNA double helix and/or distribution of curvature along the DNA molecule: PATH and MAP options. In what follows we refer to the MAP option of the software, which may be considered as a procedure of transformation of strings into numerical sequences: every substring of an input DNA sequence of a predefined size is converted into a numerical value that is a predicted DNA curvature value. Many details related to the phenomenon of the intrinsic DNA curvature and to the method of its prediction are compiled in Appendix C of this manuscript.

The MAP option of the software CURVATURE is used to predict the distribution of curvature along any DNA sequence. This program takes, as input, the DNA sequence and calculates the likely degree of curvature at each point along the molecule using a sliding window with a predefined size. A curvature value at position $i$corresponds to a curvature of the arc in approximation to the predicted DNA path. The arc approximates a segment of the predefined size with a center of the segment at position i. Details are given in Appendix C.

To describe any profile construction procedure (see Section 1 in this chapter), the following parameters should be introduced: (i) text T; (ii) a set of M specified positions $\{m_i\}$; (iii) the size of the sliding window w; (iv) interval [-a, b]; and (v) function f. Here are the descriptions of these parameters, which are suitable for the clustering of several prokaryotic genomes (the descriptions here are analogous to those introduced above for the construction of eukaryotic LC profiles).

1. Text T represents the sequence of one complete prokaryotic genome, for example, *Campylobacter jejuni* from the GenBank. A description of a genome in the GenBank contains a lot of information on coding genes (see an example in Table 7.5, where some data regarding loci 0702-0706 of *C. jejuni* are presented):
2. A set of markers is the set of all suitable start codon positions of the selected complete prokaryotic genome (see prokaryotic gene structure and

the related terminology in Appendix C). In our study of curvature distribution around the 5'-ends of the CDS, we processed only CDS longer than 125 nucleotides and flanked by upstream intergenic regions longer than 125 nucleotides. In the abovementioned example (Table 7.5) the markers 253642 and 255090 of the loci Cj8421_0703 and Cj8421_0705 are not included in the set because of the overlap with previous CDS; the marker 254377 of the locus tag Cj8421_0704 is not included in the set because of the short intergenic region; the marker 256521 of the locus tag Cj8421_0706 is included in the set.

3. The size of the sliding window is selected according to biological considerations (see Appendix C). In [23, 155] it is equal to 150 bp (an alternative window size of 63 bp was tested as well and was found less suitable). In [155] a slightly different size of 120 bp is used.
4. More than a single variant of the parameters *a* and *b* for the interval [-a, b] were used. For curvature profiles used in [23, 155] the interval is defined as [-500, +500]; in Kozobay-Avraham *et al.* [155] the interval is [-400, 400]. The authors aimed to take a neighborhood [-a, b] around the start of CDS. However, the entire regions ±500 bases' length around the starts of translation were used exclusively in cases where all 500 nucleotides upstream were located in an intergenic region and all 500 bases downstream belonged to CDS. (The same statement is true regarding the interval [-400, 400] used in [156, 155] and the interval [-200, 200] used in [156].) Otherwise, only relevant downstream coding or upstream noncoding pieces were used. For example, the considered neighborhood of the start of the gene Cj8421_0706 /256521..256979/used in [23, 155] is [-274, 458], because 256521-256247=274, and 256979-256521=458 (see Table 7.5).
5. Two modifications of DNA curvature profiles have been used: curvature profiles, based on absolute DNA curvature values [23, 155], and curvature excess profiles [156].

Table 7.5 Excerpt of the annotation of *Campylobacter jejuni subsp.* jejuni CG8421.

CDS	252792..253655 /locus_tag="Cj8421_0702"
CDS	253642..254367 /locus_tag="Cj8421_0703"
CDS	254377..255093 /locus_tag="Cj8421_0704"
CDS	255090..256247 /locus_tag="Cj8421_0705"
CDS	256521..256979 /locus_tag="Cj8421_0706"

7.3.2 Construction of Genomic DNA Curvature Profiles

In [23, 155] curvature distributions around the 5'-ends of the CDS of 21 genomes were studied (see Fig.7.5). The expectation was that UCS would be located upstream to starts of translation in regions of putative promoters. In Fig.7.5 all curves are shown on the same scale but shifted along the Y-axis for illustration purposes. The curvature values are presented in nucleosome units. Major ticks on the Y-axis correspond to the curvature of 0.02 nucleosome unit. The clustering of the genomes led to a division into three groups: predominance of big mesophilic genomes in the first group (Fig.7.5a), small mesophilic genomes in the second group (Fig. 7.5b), and all hyperthermophilic genomes (Fig.7.5c).

In the study based on a very small amount of complete prokaryotic genomes there were two major statements:

1. Distribution analysis showed a substantial presence of more curved pieces, 100-200 bases upstream to the start of CDS in such mesophilic genomes as *Escherichia coli, Mycobacterium tuberculosis, Bacillus sp., Synechocystis, Haemophilus influenzae,* and *Helicobacter pylori.*
2. By contrast, both euryarchaeal and bacterial hyperthermophilic species did not demonstrate such a property. DNA curvature probably does not play any significant biological role in gene regulation of hyperthermophilic species.

The concluding remarks in [23, 155] contained the prediction that analysis of new complete prokaryotic genomes, as a rule, will show patterns of curvature distribution dependent upon normal growth temperatures.

In the following study [155] the hypothesis that the DNA curvature plays a biological role in gene regulation in mesophilic as compared to hyperthermophilic prokaryotes was verified. In Kozobay-Avraham's study [155] 105 Bacterial genomes and 16 Archaeal genomes were analyzed (compared with six euryarchaeal species and 15 bacterial in [23, 155]. Another hypothesis was checked as well: whether a genomic dinucleotide composition completely determines average curvature. Control genomes with the same dinucleotide composition for genomic and intergenic sequences were constructed separately. This was done to test the significance of results and to compare the properties of natural and artificial genomes. The construction procedure consisted of three steps: a) a genome was cut in separate genic and intergenic pieces at every 5' and 3' gene junction; b) each piece was separately reshuffled preserving dinucleotide composition; c) all the pieces were reassembled in the original order. For every genome, 10 randomized control genomes, using the abovementioned procedure of shuffling and rejoining randomly reshuffled pieces, were prepared. Magnitude and standard deviation of curvature in coding and noncoding sequences of artificial genomes by assuming and averaging 10 randomized shuffled genomes were estimated. Using a comparison

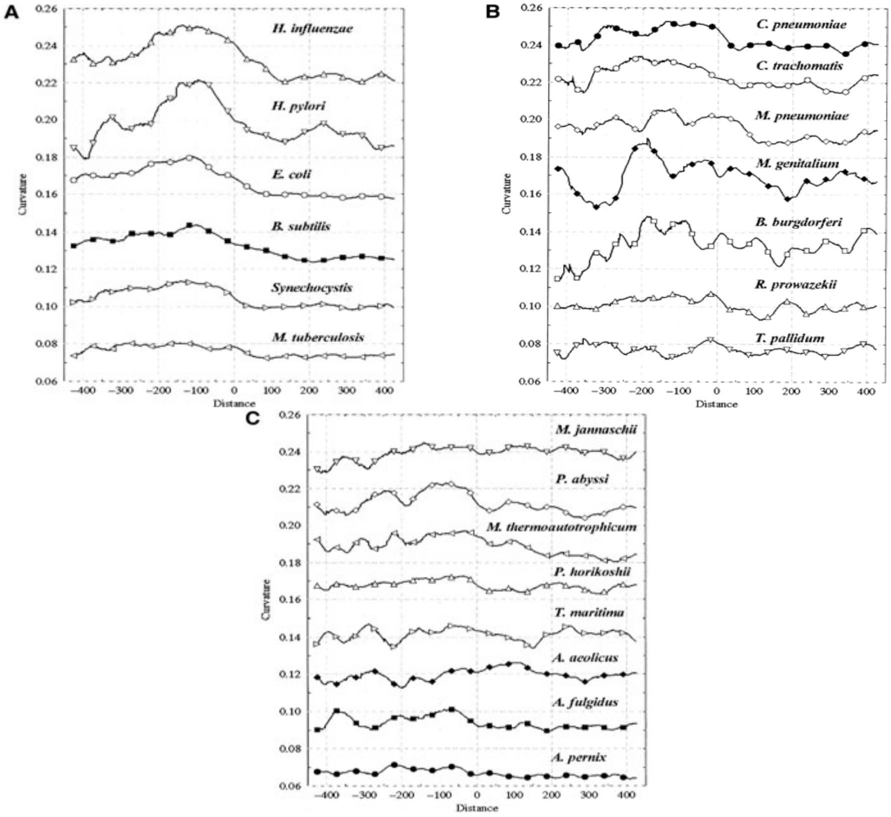


Fig. 7.5 Curvature distributions in the neighborhood of the starts of translation. For each genome, the sets of region 500 bases in length around the starts of translation were compiled. All graphs are on the same scale; for better presentation some of them were shifted along the Y-axis to prevent overlapping. Major ticks on the Y-axis correspond to the curvature of 0.02 nucleosome unit. The number of processed CDS fragments is shown in brackets. Every 50th point is marked by a corresponding symbol: circle, diamond, square, or triangle. **A. Big Mesophilic Bacteria** Haemophilus influenzae triangles up (503 CDS); *Helicobacter pylori* triangles down (357 CDS); *Escherichia coli* circles (1522 CDS); *Bacillus subtilis* squares (1439 CDS); *Synechocystis sp.* triangles right (1343 CDS); *Mycobacterium tuberculosis* triangles left (1078 CDS). **B. Small Mesophilic Bacteria** *Chlamydia pneumonia* solid circles (384 CDS); *Chlamydia trachomatis* empty circles (326 CDS); *Mycoplasma pneumoniae* empty diamonds (210 CDS); *Mycoplasma genitalium* solid diamonds (73 CDS); *Borrelia burgdorferi* empty circles (137 CDS); *Rickettsia prowazekii* triangles down (342 CDS); *Treponema pallidum* - triangles up (176 CDS) **C. Hyperthermophilic Archaea and Bacteria** *Methanococcus jannaschii* triangles down (449 CDS); *Pyrococcus abyssi* empty diamonds (307 CDS); *Pyrococcus horikoshii* triangles up (473 CDS); *Methanobacterium thermoautotrophicum* triangles left (360 CDS); *Thermotoga maritima* triangles right (233 CDS); *Archaeoglobus fulgidus* solid squares (374 CDS); *Aquifex aeolicus* solid diamonds (242 CDS); *Aeropyrum pernix* solid circles (749 CDS).

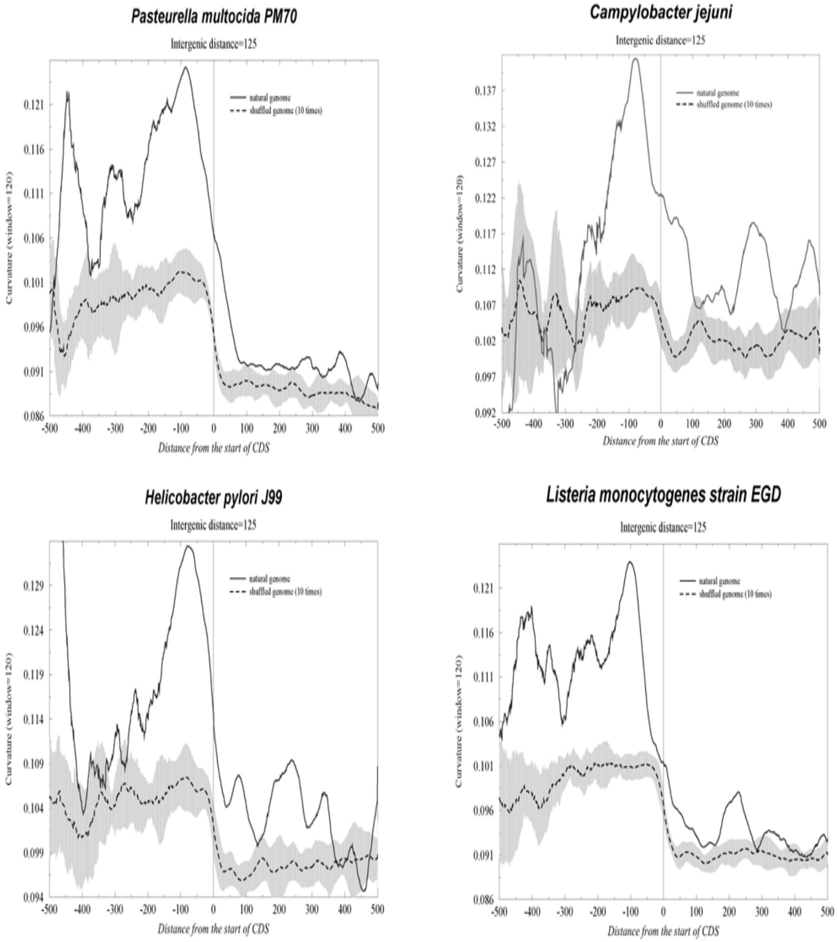


Fig. 7.6 Curvature distributions of big mesophilic AT-rich genomes e.g., *Pasteurella multocida* (757), *Campylobacter jejuni* (251), *Helicobacter pylori* (417), and *Listeria monocytogenes* (1065). The mean distributions (solid line) were obtained by averaging the distributions of all fragments from the same genome. The dash line corresponds to distribution of expected random value. The standard errors were estimated by bootstrap method using 1,000 runs. For better visibility, error bars corresponding to several distances around the 5'-end are shown separately from curvature maps. Many intergenic areas are shorter than 400 bp; therefore, standard deviation is larger for intergenic positions that are more distant from the 5'-end of gene. For this reason, the far upstream peak in the natural genome, as a rule, is not significant. *P. multocida* shows the maximal curvature value of 0.125 in position -88. C. jejuni shows the maximal curvature value of 0.141 in position -74. *H. pylori* shows the maximal curvature value of 0.13 in position -73. *L. monocytogenes* shows the maximal curvature value of 0.124 in position -100.

between natural and randomized genomes, it was found that in almost every complete prokaryotic genome the noncoding sequences were more curved than their shuffled counterparts with the same dinucleotide composition. Therefore, the conclusion presented in [155] was that genome curvature distribution of a prokaryotic genome, in general, contains more information than may be expected from its dinucleotide composition. This result is clearly visible from plots of curvature distribution around starts of genes (see Fig.7.6).

The hypothesis regarding temperature influence on curved DNA distribution was proposed in [23, 155]. The results presented by Kozobay-Avraham *et al.* in 2004 [155] are in full accordance with this theory: curvature peaks were observed wherein loop formation was expected; and among the hyperthermophilic genomes, this phenomenon was not observed.

In the following studies of Kozobay-Avraham *et al.* [156, 156] curvature excess distribution around the starts and ends of the annotated coding sequences was used instead of absolute values of DNA curvature.

Curvature excess is an apparent deviation between genomic (g_i) and random (r_i) curvature values measured in standard deviation units and calculated as follows:

$$CE = \frac{(g_i - r_i)}{\sigma_i}$$

In order to calculate random curvature value, randomized control genomes were constructed. The randomization was done by genome reshuffling without any change in dinucleotide composition. The procedure described below is different from standard procedures of genome reshuffling. In standard procedures complete genome sequences are reshuffled while every genic or intergenic sequence dinucleotide composition was preserved separately. This kind of reshuffling procedure was developed to keep local dinucleotide composition unchanged. The construction procedure consists of three steps: a) a genome is cut in separate genic and intergenic pieces at every 5' and 3' gene junction; b) each piece is reshuffled separately preserving dinucleotide composition, and c) all the pieces are reassembled in the original order. For every genome, 10 randomized control genomes are prepared using the above-mentioned procedure of shuffling and rejoining randomly reshuffled pieces. The magnitude of curvature of coding and noncoding sequences of artificial genomes (r_i) is estimated by averaging 10 randomized shuffled genomes.

In Fig.7.7 the curvature excess profiles of genomes with clear UCS are shown. As it was mentioned above, genomes of hyperthermophiles never have this property. *T. maritime* (Eubacteria), *P. horikoshii*, *P. abyssi,* and *A. fulgidus* (Archaea) are four representatives of a group of hyperthermophilic prokaryotes.

7.3.3 Clustering of Prokaryotic Genomes

In [156] curvature excess profiles were used for K-means clustering of 170 genomes, and in [156] 205 genomes were clustered.

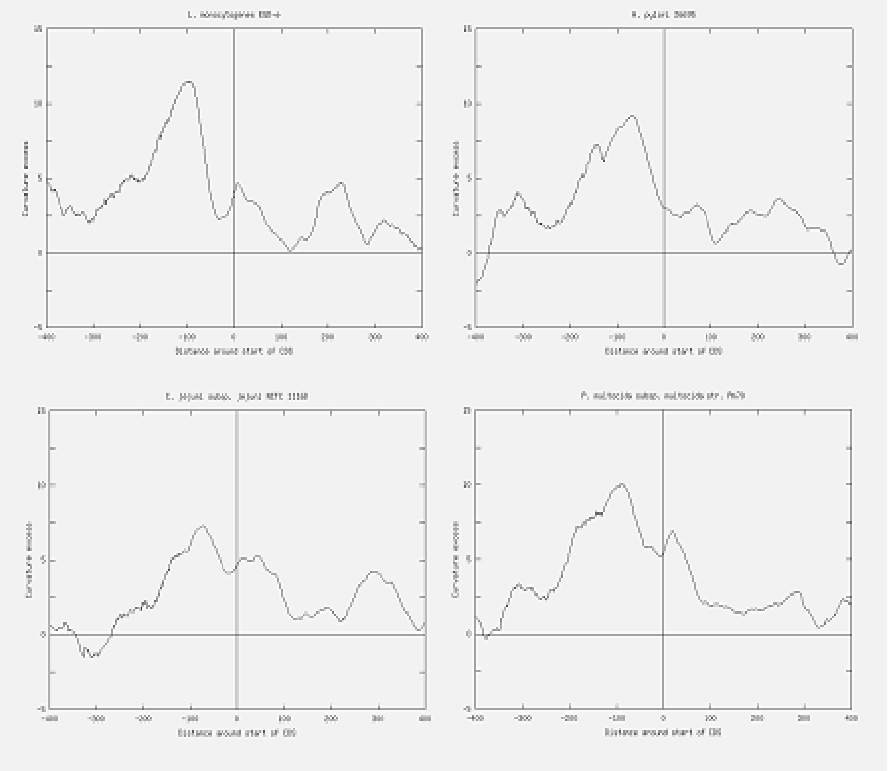


Fig. 7.7 Curvature excess profiles of the AT-rich genomes: *P. multocida*, *C. jejuni*, *H. pylori*, and *L. monocytogenes* in the neighborhood of the starts of coding sequences. The curvature excess plots correspond to the curvature plots presented in Fig.7.6. *P. multocida* shows the maximal curvature excess value of +10 in position -88. *C. jejuni* shows the maximal curvature excess value of +7.4 in position -74. *H. pylori* shows the maximal curvature excess value of +9.0 in position -73. *L. monocytogenes* shows the maximal curvature excess value of 11.5 in position -100.

7.3.3.1 Measuring a Distance between Profiles for Further Clustering

In [156] three comparison parameters of Curvature Excess (CE) were calculated: Maximal Curvature Excess (MCE) and Upstream Integral Excess (UIE) – corresponding distances d_{MCE} and d_{UIE} based on MCE and UIE were used for further clustering analyses.

The parameter MCE was determined by detecting maximal CE value. A corresponding distance d_{MCE} between two genomes is an absolute value of the difference between two maximal CE values. The distance d_{UIE} between two genomes is an absolute value of the difference between two average CE

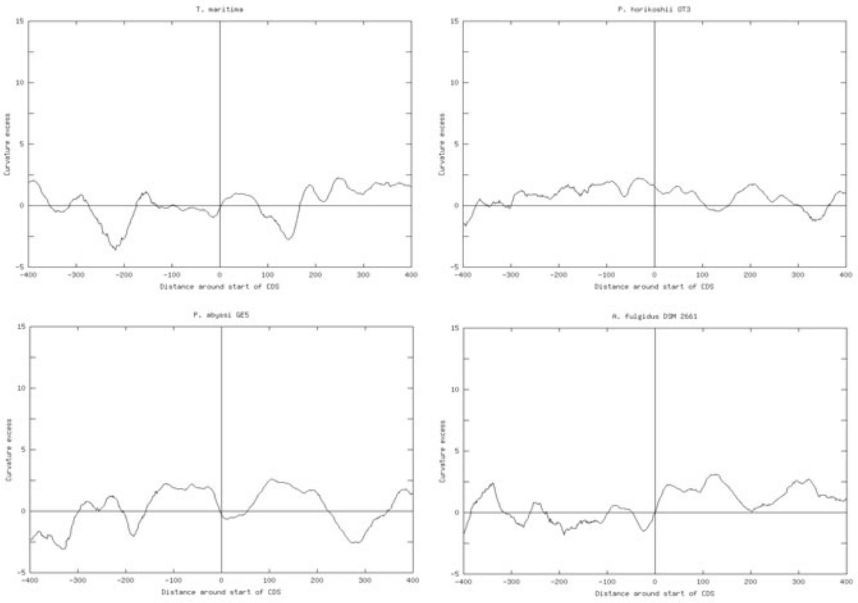


Fig. 7.8 Curvature excess profiles of the representative hyperthermophilic genomes

values. UIE represents the average in CE over X base pairs upstream to a start of translation:

$$UIE = \frac{\sum_{i=1}^{i=X} \frac{(g_i - r_i)}{\sigma_i}}{X}$$

X in the parameter UIE was determined as 125 base pairs (between –63 and –188 nucleotides) upstream to the 5'-ends of CDS, wherein promoters are usually located.

In [154] and [156] all profile data were used. In order to evaluate which of the known distance calculations is the most suitable for our data, the following distances were calculated between curvature excess profiles in promoter regions: the squared Euclidean distance, the Manhattan distance, the max distance, and three correlation distances of Spearman, Pearson, and Kendall (see, Appendix A). Partitions have been provided by means of the PAM algorithm using all of these distributions and by means of the k-means algorithm using the squared Euclidean distance. The results clearly point to the last approach as the most appropriate. Therefore, all further cluster analyses [154, 156] were performed using this distance.

7.3.3.2 Cluster Analysis Based on d$_{UIE}$ Distance

K-means algorithm over 3 clusters using the distances d_{MCE} and d_{UIE} was applied. Cluster analysis showed that the data variations in the clusters are

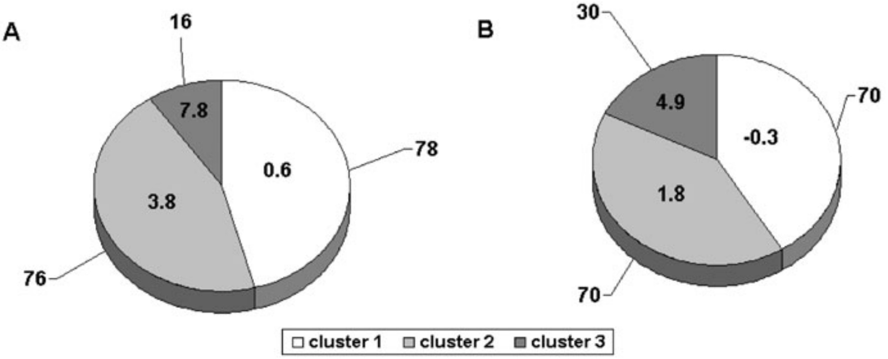


Fig. 7.9 Cluster analysis. K-means algorithm over 3 clusters was carried out using the parameter Integral Excess in (a) promoter (UIE) and (b) terminator (DIE) regions of all 170 genomes. The numbers inside each of the pie's pieces represent the mean value of curvature excess of each cluster. An amount of the genomes in each cluster is indicated outside of the pies pieces.

smaller with the UIE parameter than with those compared with the MCE parameter. This result indicates that clustering based on the UIE parameter is less biased and more reliable than the MCE. The mean values of UIE and the amount of genomes in every cluster are summarized in Fig.7.9.

In [156] the authors tried to verify previous qualitative findings regarding the factors that influence curvature distribution in promoter regions of a quantitative manner.

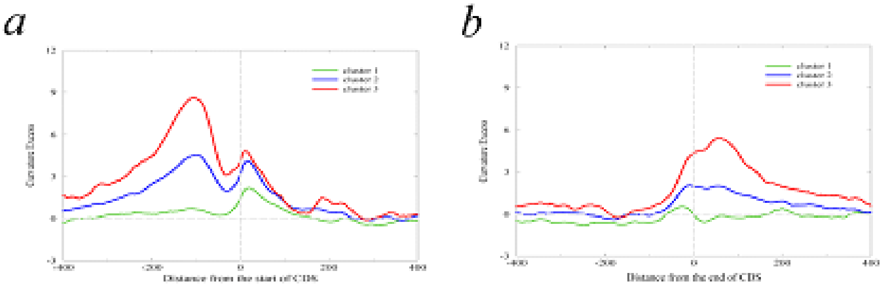


Fig. 7.10 Mean profiles (centroids) of clusters. In (1) profiles were used for K-means clustering of 170 genomes. Genomic profiles based on curvature excess distribution in the neighborhood of the starts (a) and ends (b) of genes. The centroids are obtained by averaging all profiles related to each of the three clusters obtained by K-means algorithm. The y-axis represents the curvature excess in standard deviation units and the x-axis represents the position around the start or end of translation. The highest profiles are related to cluster 3, and the lowest profiles correspond to cluster 1.

From the data obtained by cluster analyses, three histograms of UIE curvature excess were constructed (Fig.7.11). Each histogram is colored according to one characteristic: A) growth temperature, B) genome size, and C) A+T composition. Each pie plot presents the distribution of one characteristic in a particular cluster.

Cluster 3 in Fig.7.11 has the highest mean values and includes only mesophilic bacteria that have a relatively ‘big’ genome size (over 1.4 Mbp); all of the genomes in this cluster are AT-rich (above 50% A+T content in the noncoding region).

In Cluster 2, 75 out of the 76 organisms are mesophiles and 1 is a thermophile. In this cluster, about 70% are ‘big AT-rich’ and the rest are either GC-rich or ‘small’. Cluster 1, which represents the lowest mean value of curvature excesses in promoter regions, contains the lowest percentage of mesophiles compared to other clusters. The data based on the UIE show that this cluster contains all the hyperthermophiles and 99% percent of the thermophiles. Moreover, among the 51 mesophiles only 7 are ‘big AT-rich’.

Clusters 1 and 2 based on clustering, using curvature excess in terminator regions (DIE), consist of 70 genomes each. While examining these clusters, the most prominent difference from the clustering based on promoters is the distribution of the thermophilic and hyperthermophilic genomes in the clusters. This group shows homogeneous distribution: 12 genomes were included in cluster 1 (which represents the lowest mean value) and 9 genomes were included in cluster 2 (the medium). Another difference can be seen in the distribution of the ‘small’ mesophilic genomes. While performing cluster analysis on the promoters’ curvature (UIE), cluster 1 includes smaller mesophilic genomes than cluster 2, which corresponds with the expectations mentioned in Kozobay-Avraham *et al.* (2004). Performing the same analysis on terminators (DIE), it was found that the relationship between genome size and curvature in terminator regions is weaker than in promoter regions.

Another interesting picture can be seen when examining the phyla distribution at every cluster. Clustering based on UIE curvature forms cluster 3 with 14 genomes out of 16 belonging to three groups that were known to be AT-rich: *Firmicutes* and *Proteobacteria* gamma and epsilon subdivisions, which are more AT-rich than the other members of the *Proteobacteria* phyla. (A similar picture can be seen using the parameter DIE; about 70% of the genomes presented in cluster 3 belong to these phyla.) At clusters 2 and 1, based on the parameter UIE, the picture is more heterogenic. However, some interesting points can still be drawn, i.e., all seven members of the phylum *Chlamydia* belong to cluster 1. In this phylum, the variation in size is relatively very small - from about 1 to 2.4 Mbp. On the contrary, if we look at all seven genomes of *Cyanobacteria* wherein the variation of size is very big - from about 1.6 to 6.4, or the 13 genomes of the *Actinobacteria* - from 0.9 to 9 Mbp, the cluster distribution is less homogeneous. This phenomenon indicates a relationship between the size of the genomes and clustering, based on curvature excess at the promoter regions. As was expected, all members

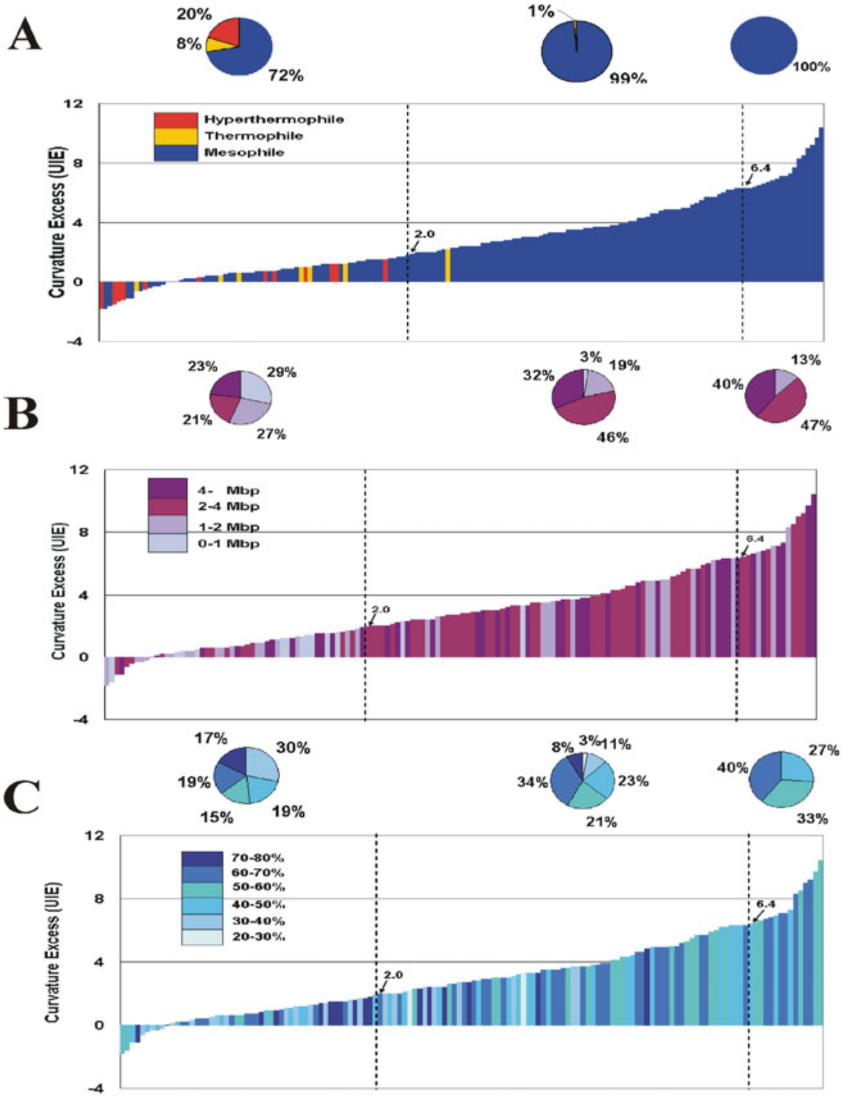


Fig. 7.11 Histograms of curvature excess in promoter regions. The coloring of each histogram represents distributions of one characteristic along the clusters: (A) optimal growth temperature (OGT), (B) genome size, and (C) A+T composition. The parameter UIE, which was calculated for the purely upstream window (from nucleotide -63 to 188), was used to build the histograms. In histogram A all of the genomes are represented, excluding seven genomes with unknown OGT. In histograms B and C only mesophilic genomes are represented. Curvature excess threshold of each cluster is indicated on the left side of the vertical dashed line. Above each histogram three pie plots are presented for better visibility of the character distribution in each cluster

of the kingdom Archaea, except for two mesophilic *Methanosarcina* strains: *M. acetivorans* and *M. mazei* appear in cluster 1, which was characterized by the lowest mean UIE value.

The discussion on these findings is placed in Appendix C. Here we recount a shorter version of this discussion.

In publication [155] it was shown that the most prominent phenomenon is that an abundance of UCS in a genome was determined by the temperature of its habitat. Other characteristics, such as genome size and A+T composition, also influence this phenomenon. When growth temperature, genome size, and A+T composition are taken into consideration – peculiar observations influencing curvature at regulation sites can be explained. For example, there was no representation of very AT-rich (above 70%) genomes found in clusters with higher mean values, but underlying genome size revealed that most of these are 'small' genomes.

7.3.3.3 Cluster Analysis Based on the Squared Euclidean Distance

In this section we describe an approach and results of the study [156] of Kozobay-Avraham textitet al. (2009). The genomic database of [156] consists of 205 prokaryotic genomes. The following genomic characteristics were gathered from genomic annotations and from the literature: optimal growth temperature, genome size, $A + T$ composition, and taxonomic description.

Optimal growth temperature - the organisms belong to four temperature groups, as defined in the literature (see Chapter 2).

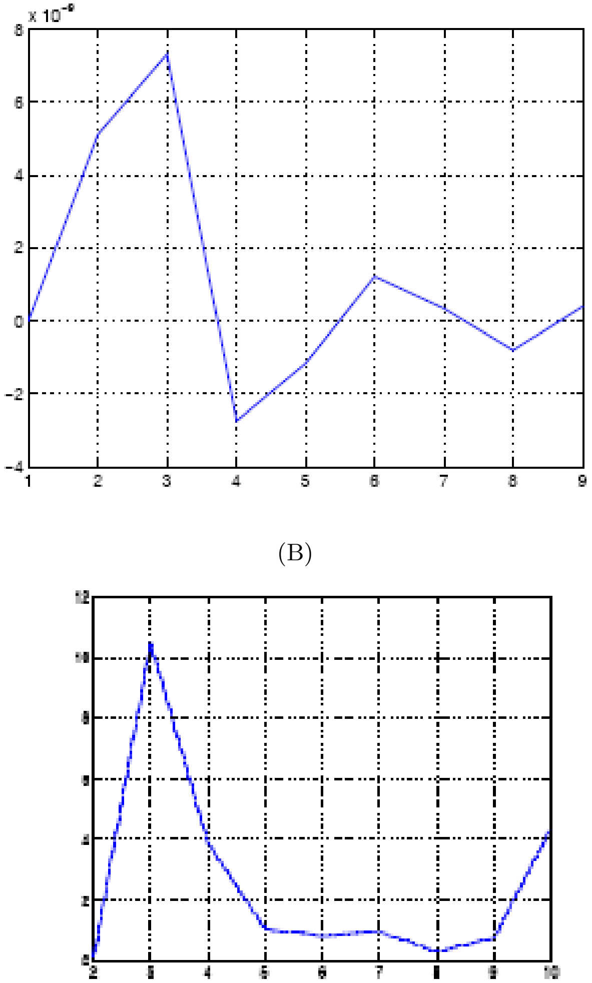
Taxonomy - prokaryotes fall into one of two groups, Archaebacteria (ancient forms thought to have evolved separately from other bacteria) and Eubacteria. Archaebacteria, sometimes called Archaea, emerged at least 3.5 billion years ago and lived in environments that existed when the earth was young. Many hyperthermophiles belong to Archaea. The genomic database of [156] consists of 23 Archaeal and 182 Bacterial genomes. Among the Archaeal genomes 19 representatives are thermophiles or hyperthermophiles; four genomes are mesophiles.

Genome sizes - prokaryotes have relatively small genomes: from very short genomes with lengths less than 1Mb - up to about 9Mb. The size of a genome is a relevant factor because the smallest genome-sized prokaryotic species, the obligate endocellular parasites, when compared to their free-living relatives, have preferentially lost many regulatory elements, including many transcriptional factors.

$A + T$ composition - this feature is relevant to our analysis because the magnitude of DNA curvature depends on it. In AT-rich segments curved DNA fragments occur more frequently than in AT-poor segments. Moreover, it was shown that strongly curved DNA fragments must possess high A+T content.

The cluster analyses were performed using the k-means and the PAM methods. Two cluster stability indexes were applied: Krzanowski and Lai index ([158]) and Sugar and James index ([241]). The graphs (Fig. 7.12) of these two cluster stability indices (see Appendix A, Clustering methods section) demonstrate that both indices provide evidence that the true number of clusters is three. The possible number of clusters is presented on the X-axis. On the Y-axis, the values of an appropriate index are plotted. From this point on in the text the number of clusters is three.

(A)



(B)

Fig. 7.12 The cluster stability indices. Graphs of the Sugar and James index (A) and Krzanowski and Lai index (B).

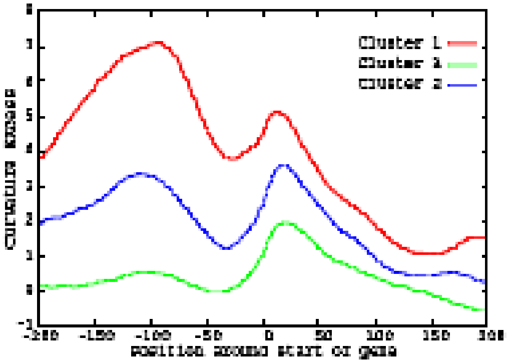


Fig. 7.13 Mean profiles (centroids) of clusters. In Kozobay-Avraham et al. (2009) profiles were used for K-means clustering of 205 genomes. Genomic profiles based on curvature excess distribution in the neighborhood of the starts of genes. The centroids are obtained by averaging all profiles related to each of the three clusters obtained by the K-means algorithm. The y-axis represents the curvature excess in standard deviation units, and the x-axis represents the position around the start of translation. The highest profiles are related to cluster 1, and the lowest profiles correspond to cluster 3.

Three clusters were obtained using k-means algorithm and the graphs present centroid profiles related to each of three clusters. The highest profile is related to cluster 1, and the lowest profile corresponds to cluster 3. Cluster 1, the smallest cluster containing genomes with the highest curvature excess values in promoter regions is rather homogeneous. The cluster contains exclusively mesophilic prokaryotes that have genome sizes larger than 1.4 Mb and a high A + T composition. The PAM method gives rather similar results; 92% genomes in the smallest cluster are also big AT-rich mesophilic genomes.

Table 7.6 presents the results of clustering obtained by the K-means algorithm cross-tabulated with temperature classifications. The FM correlation coefficient is equal to 0.48 for the correlation between temperature and curvature-based partitions. Cluster 1 contains only mesophilic prokaryotes; all four psychrophiles are located in cluster 2; a majority of the thermophiles

Table 7.6 Cross-tabulation count between temperature and clusters

Cluster \ Temperature	Psychrophiles	Mesophiles	Thermophiles	Hyperthermophiles
1	0	48	0	0
2	4	67	4	4
3	0	57	8	13

and hyperthermophiles is located in cluster 3. Mesophilic genomes have notable representation in all clusters.

The distribution of Archaea among three clusters is not shown but is not surprising because the majority of the processed Archaea is hyperthermophiles, and hyperthermophiles are mainly Archaea. Therefore, the distribution of archaeal genomes is similar to the above mentioned distribution of hyperthermophiles. Indeed, it should be mentioned that mesophilic Archaea are grouped together with mesophilic Eubacteria, while hyperthermophilic Eubacteria are grouped together with hyperthermophilic Archaebacteria.

Genome size and A + T composition have already been found to have influence on curvature distribution [156]. In [156] the genomes were arbitrarily divided into two groups with a threshold of 1.4 Mbp. Lengths of the prokaryotic genomes processed in [156] range in size from 490,885 to 9,105,828 bp. In order to verify the previous intuitively selected threshold of 1,400,000 bp, 2 *t*-tests were performed by Kozobay-Avraham *et al.* [156]: one according to the median (2.4 Mbp) and the second according to the arbitrary value (1.4 Mbp). It was found that the differences between the mean values of the groups were significantly higher when the threshold of 1.4 Mbp was used. Thus, these results verified the intuitive threshold between “small” and “big” genomes used in previous publications [156, 155]. Correlation of A + T composition with clustering described in [156, 155] was verified in Kozobay:2009c as well.

7.3.4 Sub-clustering of Coding Regions after Clustering Based on the Squared Euclidean Distance

Centroids of the three clusters in the neighborhood of the starts of genes (from -200 bases to +200 bases) are shown in Fig. 7.13. The results of the clustering were based on measuring squared Euclidian distance d_{ij} between profiles x_i and x_j in the upstream region only:

$$d(x,y) = \|x-y\|^2 \Rightarrow d_{ij} = \sum_{l=-200}^{0} \left(x_i^l - x_j^l\right)^2$$

As a further step, sub-partitions for each one of the obtained clusters were considered separately for the purpose of achieving a more detailed biological interpretation. The k-means algorithm, accompanied by the Euclidian distance with the number of clusters equal to 3, has been used. The centroids of all nine clusters are presented in Fig.7.14.

Several surprising features of curvature excess distributions in the coding regions were observed. One of them is a sharp maximum immediately after the start of translation typical for sub-clusters 1,1 and 1,2; 2,2 and 2,3; and 3,2. It is worthwhile to remind the reader that the curves are of a curvature excess – this means that the presence of a sharp maximum of normalized

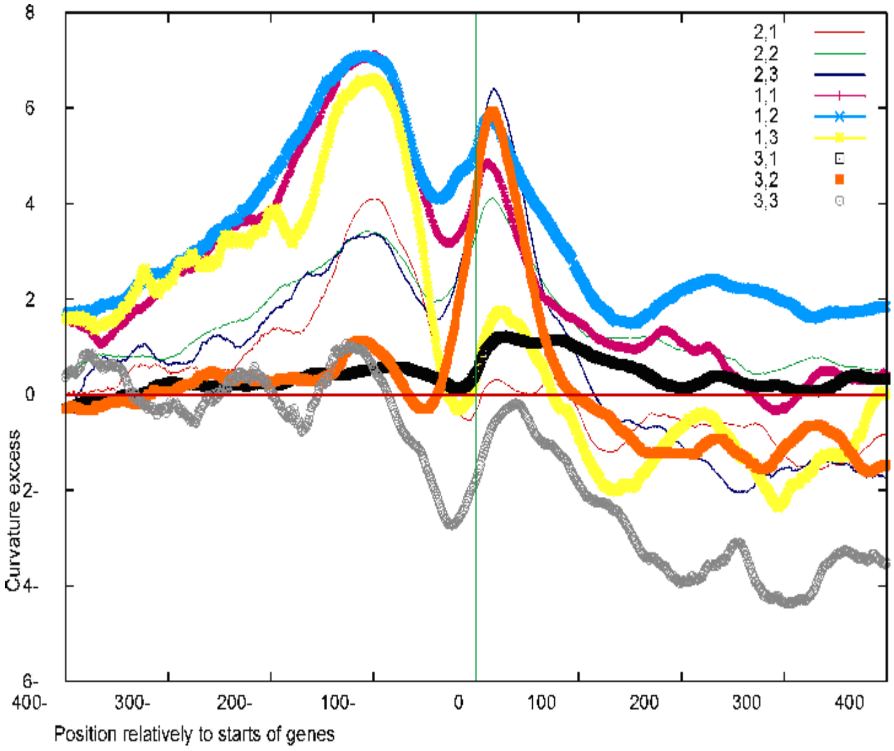


Fig. 7.14 Mean profiles (centroids) of sub-clusters from Kozobay-Avraham *et al.* (2009). The sub-clusters are notated with accordance to Fig.7.13: the first digit from 1 to 3 is an index of a cluster, and the second digit is an index of a sub-cluster in it.

Table 7.7 Correlation between taxonomy and G5MCC

Phylum	**Class**	**Genomes with G5MCC**	**Genomes without G5MCC**
Actinobacteria	Actinobacteria	12	4
Chlamydiae	Chlamydiae	1	6
Cyanobacteria		2	5
Firmicutes	Bacilli	22	8
Firmicutes	Mollicutes	2	8
Proteobacteria	Alphaproteobacteria	15	5
Proteobacteria	Betaproteobacteria	10	0
Proteobacteria	Gammaproteobacteria	37	7
Proteobacteria	Deltaproteobacteria	4	2
Proteobacteria	Epsilonproteobacteria	6	0
Spirochaetes	Spirochaetes	1	7

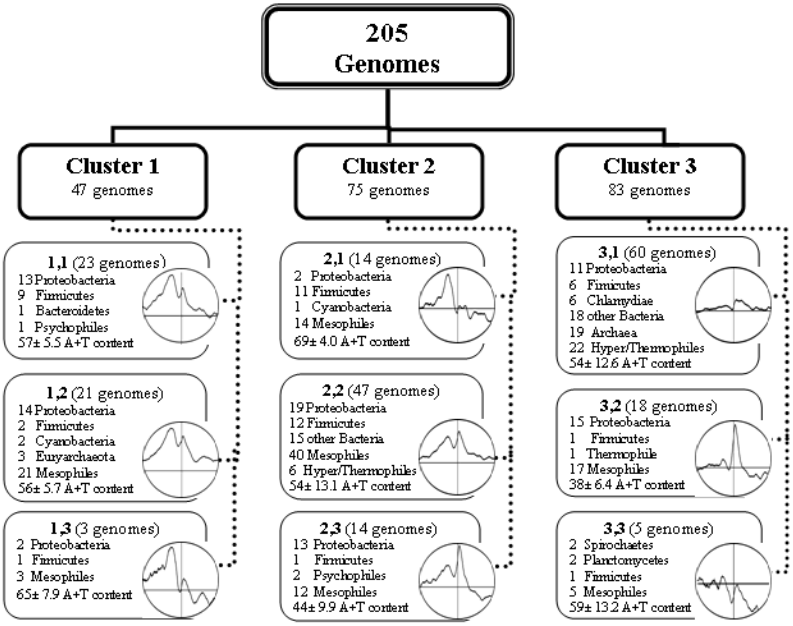


Fig. 7.15 A diagram of clusters and sub-clusters. Cluster notation is as it appeared in Fig.7.15, Table 7.7 Interestingly, A+T composition of G5MCC is rather heterogeneous.

curvature values at a certain location, does not necessarily lead to the presence of a maximum of absolute curvature values at the same location. Let us denote this group as "Genic 5'-end Maximum Characterized Clusters" or G5MCC. The most surprising among these profiles is the sub-cluster 3,2, which belongs to cluster number 3 with the lowest curvature excess values in promoter regions. Sub-cluster 3,2 presents the sharpest inclination of curvature excess profile immediately after the start of genes. Let us take notice that G5MCCs are usually mesophiles: all thermophiles (including hyperthermophiles) are located in two of nine clusters - 22 thermophiles belong to the cluster 3,1, which is not G5MCC, but 6 thermophiles also belong to cluster 2,2 (*Geobacillus kaustophilus*, *Streptococcus thermophilus, Thermoanaerobacter tengcongensis,* and *Thermobifida fusca* - are thermophiles; *Carboxydothermus hydrogenoformans* and *Pyrococcus horikoshii* - are hyperthermophiles).

The profile of the sub-cluster 3 of cluster 1 with the highest curvature excess values in promoter regions has a contrasting surprising feature: a sharp descent after the start of genes. However, this group contained only three genomes; therefore, to explain this phenomenon further investigation is needed.

A few questions are still unanswered. Why was a surprisingly high DNA curvature located immediately after the start of the gene? Why would evolution frequently keep an extensive curvature at the 5'-ends of many protein-coding sequences? Why is the phenomenon of G5MCC so typical for many representatives of proteobacteria but seemingly a rather rare episode for Chlamydiae or Spirochaetes? Here, we can propose some speculations.

The absence of excessive curvature in promoter regions of almost all "small" genomes probably reflects an adaptive selection. Small genomes are usually obligate endocellular parasites that evolutionarily adapted to utilize their genome host and consequentially lost nonfunctional sequences, such as regulatory elements. Similar to our findings related to regulation regions, a genome's size seems to have influence on the curvature distribution in coding regions as well.

7.3.5 Conclusions Pertaining to Prokaryotic Species Classification Based on DNA Curvature Distribution

The main factors influencing curvature distribution in promoter regions of the prokaryotes were found, in order of importance, as: optimal growth temperature, genome size, and A+T composition. The possible combinations among these factors can explain, for example, the homogenous distribution of mesophilic genomes along clusters with high, moderate, and low curvature excess. The absence of excessive curvature in almost all thermophiles and hyperthermophiles brings them all together in a mutual cluster of low curvature excess, while the majority of the mesophilic AT-rich genomes are located in other clusters. Clusters containing genomes with the highest curvature excess values in promoter regions contain exclusively mesophilic prokaryotes that have big AT rich genomes.

Chapter 8
Genome as a Bag of Genes – The Whole-Genome Phylogenetics

In Chapter 3, we gave a short description of the Bag-of-Words and the Bag-of-Tokens models. We also provided a few examples of the approach which, employs, instead of meaningful words, strings of letters, which needn't have any definite sense (the so-called N-grams). In the present chapter, another application of the Bag-of-Tokens model is described, - this time, the token being a member of a set of genes Besides, as a token property, the gene length is used instead of the token frequency,

8.1 Background

One of the main applications of the N-gram-based methods is comparison of long genetic sequences, in particular, whole genomes. Such comparison requires neither preliminary nor a posteriori alignments to reveal homologous fragments in both sequences, being, therefore, also refered to as alignment-free sequence comparison [272], [21]. An obvious advantage of the N-gram frequency-based approach is its simplicity as compared to other computer-intensive and operationally-complex techniques.

However, the methods based on the calculation of N-gram occurrences cannot replace the methods based on the discovery and investigation of homologies. In this section, we present a method based on the annotations of gene of whole genomes. This method is especially applicable to prokaryotic taxonomy due to the fact that protein-coding material comprises the major portion of a prokaryotic genome (see Chapter 1). It is shown in the present chapter that the result of the genome classification produced by this method is, actually, a phylogenetic tree (see Chapter 2 for the definitions). According to Woese and colleagues [285], the phylogeny which is based on the molecular evolution data is usually obtained by the comparison of highly conserved genes. In the framework of this approach, which was introduced by Zuckerkandl and Pauling ([295]), one gene family (usually rRNA genes) is chosen to represent whole genomes, so the distance between two genomes is defined as the distance between the appropriate representative genes.

A. Bolshoy et al.: Genome Clustering, SCI 286, pp. 147–160, 2010.
springerlink.com

Although the above approach led to a great progress in molecular evolution, the drawbacks of establishing a phylogeny on the basis of any single-gene family are also well known. For instance, because of functional limitations imposed on the number of possible mutations, there exists a phenomenon of saturation, which results in a non-linear transformation of the distance scale. Species phylogenies derived from the comparison of single genes are rarely consistent with each other, in particular, with respect to horizontal gene transfer [57], [238], [9], [201]. Another well-known problem associated with the above approach relates to the possibility that the evolutionary history of any single gene differs from that of the whole organism.

As more and more genomes are being completely sequenced, phylogenetic analysis enters a new era - the era of phylogenomics. Recent studies have demonstrated the power of this approach, which has the potential of giving answers to some fundamental evolutionary questions. The whole genomes of living organisms provide a large body of information on their phylogenetic relationships. There are fewer approaches to deriving phylogenies on the basis of extensive genomic information than on the basis of a small number of genes [243], [239]. According to one of such approaches, a genome phylogeny is established on the basis of gene content [274], [286], [156]. Snel *et al.* [239] present a distance-based phylogeny of 13 completely sequenced genomes of unicellular species. The similarity between two species is defined as the number of common genes divided by the total number of genes. The authors propose to interpret the distance introduced in such a way in terms of evolutionary events of acquisition and loss of genes.

The research in the new field of phylogenomics - embedding of a genome into a coordinate space - is closely related to the topic of our book. In Chapter 3 we defined a vector space model as an algebraic model for representing text documents as vectors of identifiers, such as, for example, index terms; a document is represented as a vector. Each dimension corresponds to a separate term. The document set D is represented by a matrix A, in which each column stands for a document and item (i, j) stands for the frequency of term i in document j. If we replace "text document" to "genome" and "index term" to "gene" then we would get the following definition: a gene-content based model for genomes is an algebraic model for representing genomes as vectors of genes. The set of genomes D is represented as a matrix A, in which each row stands for a genome, and each column stands for a gene, and each item stands for the property of a member of a gene family i in a genome j. If the rows and columns are binary, a row of the matrix A is referred to as a *phylogenetic profile* [152], [247]. Tekaia and Yeramian [247] defined a *conservation profile of a protein* as an n-component vector of zeros and ones, which describes the protein's conservation pattern across n species. As an illustration, the authors analyzed a genome tree based on conservation profiles and found significant correspondence between the tree and the traditionally recognized taxonomies, along with a series of departures from the conventional clustering. As it was mentioned in a review of Snel *et al.* [238], the phylogenetic value of genome trees is not as commonly accepted as that of gene trees simply because different parts of a particular genome do not necessarily have the same evolutionary history. This raises the question as to the effectiveness of constructing a phylogeny at the genome level [153].

The method that was presented in [25], which we describe below, is closely related to a group of methods based on the presence and absence of genes; however, it may use the information related to different inherent properties of genes also. The method is based on a methodologically novel concept of a genome tree construction based on orthologous gene conservation profiles. Suppose we have n genomes for which we would like to construct a genome tree. We define an *orthologous gene property conservation profile* of a gene x as an n-component vector of zeros and positive values, which reflects an evolutionary conservation history of a property p across the n species. This is a methodologically novel concept of genome trees based on orthologous gene conservation profiles in multiple species. In [25] the property under consideration is the length of genes. Namely, the value of the $i-$th component of profile x equals zero, when gene x is absent from genome i; otherwise this value is equal to the ratio of an average length of the paralogs related to the family of genes of type x in genome i to the sum of all such averages over all genomes. The average lengths are calculated using the database Clusters of Orthologous Groups [214], [154]. It should be pointed out that other evolutionary properties, such as the gene GC-content or a Codon Usage Index of the gene, can also be used.

8.2 The Information Bottleneck Method

8.2.1 Clusters of Orthologous Groups

The data is chosen from the COGs of proteins with regard to the comparison of protein sequences. Bolshoy and Volkovich used the COG database in [250] and we show in this chapter the results obtained on the database of COGs [http : //www.ncbi.nlm.nih.gov/COG] that appeared to be the last COG database officially presented at the COG page. The list of all genomes and their partial taxonomy is shown in Tables 8.1 and 8.2. In the official COG database employed here, 63 sequenced prokaryotic genomes and three genomes of unicellular eukaryotes are included (38 *orders*, 28 *classes*, 14 *phyla*). An illustration to the typical COG data is presented in Table 8.3. One can see that many genomes possess orthologous genes from the COG006 but not all of them. Some genomes have only one protein from the COG, but other genomes have many paralogs. Some paralogs have very close protein lengths, while, for example, *Candida* (cda) has a very heterogeneous set of proteins.

Table 8.1 List of all unicellular organisms presented in COG database

Archaea	
Crenarchaeota	
pya	Thermoproteales - Pyrobaculum aerophilum
ape	Desulfurococcales - Aeropyrum pernix
sso	Sulfolobales - Sulfolobus solfataricus
Euryarchaeota	
afu	Archaeoglobales - Archaeoglobus fulgidus
hbs	Halobacteriales - Halobacterium sp. NRC-1
mth	Methanobacteriales - Methanothermobacter thermautotrophicus
mja	Methanococcales - Methanococcus jannaschii
pab	Thermococcales - Pyrococcus abyssi
pho	Pyrococcus horikoshii
tac	Thermoplasmales - Thermoplasma acidophilum -
tvo	Thermoplasma volcanium
mka	Methanopyrales - Methanopyrus kandleri AV19
mac	Methanosarcinales - Methanosarcina acetivorans str.C2A
Bacteria	
aae	Aquificales - Aquifex aeolicus
tma	Thermotogales - Thermotoga maritima
fnu	Fusobacterales - Fusobacterium nucleatum
bbu	Spirochaetales - Borrelia burgdorferi
tpa	Treponema pallidum
dra	Thermus/Deinococcus group - Deinococcus radiodurans
ctr	Chlamydiales - Chlamydia trachomatis
cpn	Chlamydophila pneumoniae CWL029
	Firmicutes
Bacillales	
bsu	Bacillus subtilis
bha	Bacillus halodurans
lin	Listeria innocua
sau	Staphylococcus aureus N315 Lactobacillales
lla	Lactococcus lactis
spy	Streptococcus pyogenes
spn	Streptococcus pneumoniae TIGR4
Mycoplasmataceae	
mge	Mycoplasma genitalium
mpn	Mycoplasma pneumoniae
mpu	Mycoplasma pulmonis
uur	Ureaplasma urealyticum
cac	Clostridiales - Clostridium acetobutylicum
Actinobacteria	
mtu	Mycobacterium tuberculosis H37Rv
mtc	Mycobacterium tuberculosis CDC1551
mle	Mycobacterium leprae
cgl	Corynebacterium glutamicum

Table 8.2 List of all unicellular organisms presented in COG database (continue)

Proteobacteria	
pae	Pseudomonadales - Pseudomonas aeruginosa
eco	Enterobacteriales - Escherichia coli K12
ecs	Escherichia coli O157:H7
ecz	Escherichia coli K12 O157:H7 EDL933
buc	- Buchnera sp. APS
ype	- Yersinia pestis
sty	- Salmonella typhimurium LT2
xfa	Xanthomonadales - Xylella fastidiosa 9a5c
vch	Vibrionales - Vibrio cholerae
hin	Pasteurellales - Haemophilus influenzae Rd
pmu	- Pasteurella multocida
rso	Burkholderiales - Ralstonia solanacearum
nme	Neisseriales - Neisseria meningitidis NC58
nma	- Neisseria meningitidis Z2491
cje	Campylobacteriales - Campylobacter jejuni
hpy	- Helicobacter pylori 26695
jhp	- Helicobacter pylori J99
ccr	Caulobacterales - Caulobacter vibrioides
rpx	Rickettsiales - Rickettsia prowazekii
rco	- Rickettsia conorii
mlo	Rhizobiales - Mesorhizobium loti
atu	Agrobacterium tumefaciens strain C58
sme	Sinorhizobium meliloti
bme	Brucella melitensis
syn	Cyanobacteria - Synechocystis PCC6803
nos	Nostoc sp. PCC 7120
Eukaria	
sce	Ascomycota - Saccharomyces cerevisiae
spo	Schizosaccharomyces pombe
ecu	Microsporidia - Encephalitozoon-cuniculi

Here, a brief description of the COG database construction follows. COGs are constructed from the results of all-against-all BLAST [9] comparison of proteins encoded in complete genomes by detecting consistent groups of genome-specific best hits. The COG construction procedure does not rely on any preconceived phylogenetic tree of the included species with the exception of certain obviously related genomes (for example, two species of mycoplasmas or pyrococci). The related genomes are grouped prior to the analysis, to eliminate strong interrelation of the best hits. Only the gene pairs, which are conserved in three or more genomes, are considered.

Table 8.3 An example of data on gene lengths. COG0006 contains aminopeptidase proteins from different Archaeal and Eubacterial genomes. Only part of the genomes in the study is shown here. The protein lengths are measured in a number of amino acids

aful	363	ccre	603
hbsp	369 391	paer	405 444
mjan	347	ecoli	361 441 443
mthe	336	Zecoli	361 441 443
tacid	331 360	buch	
tvol	360	vibrio	597
pyro	356 351 365	hinf	430
paby	351 355 365	pmul	441
aero	349	xfa	400 446
yeast	399 511 535 749	nmen	598
cda	642 425 490 699 450	nmenA	659
aquae	354	hpyl	357
tmar	359	hpyl99	357
drad	312 349	cjej	596
mtub	375 372	rpxx	591
mlep	376	ctra	356
llast	352 362	cpneu	355
spyo	361 357	tpal	774
bsub	363 353	bbur	592
bhal	360 406 355 364	uure	357
syne	441	mpn	354
mlot	395 403 389 377 386 597 383 362 375 391	mgen	354

8.2.2 Matrix Preparation

8.2.2.1 Gene-Content Matrix

The COG collection consists of 138,458 proteins, which form 4,873 COGs and comprise 75% of the 185,505 (predicted) proteins encoded in 66 genomes of unicellular organisms. Therefore, the data on orthologs is represented a matrix of size [66:4,873], where the matrix element m_{ij} equals 1 if genome $i$encodes a protein ortholog belonging to COGj and is 0 otherwise. It is well-known that only a tiny fraction (about $\sim$1%) of the COGs are present in all 66 genomes, and even the COGs that are present in all Eubacteria or in all Archaea constitute a minority This property of our matrix makes it a sparse matrix.

8.2.2.2 Gene-Length Matrix

A gene-length matrix is a sparse matrix [66:4,873] similar to the described above gene-content matrix. The matrix element m_{ij} is the average length of the protein orthologs of genome i belonging to **COG**j. The information on orthologous genes in prokaryotic and yeast genomes was derived from the **COG**s as described in the previous approach. An element m_{ij} of the [genome, **COG**] matrix is obtained in two steps:

1. In the first stage, a cleanup procedure is applied to make the data more consistent. For this purpose, we try to eliminate outliers[1] from a COG set of gene lengths. There is no rigid mathematical definition of what constitutes an outlier; thus, we use a rather naive approach; outliers are eliminated using the well known “three-sigma” rule. The mean μ and the standard deviation σ of the lengths of each **COG**’s items were calculated and the proteins having the length less than $(\mu - 3\sigma)$ or greater than $(\mu - 3\sigma)$ were termed as “outliers”. The outliers were not counted up. In general, about 3% of all protein lengths were excluded from further calculations.
2. At the second step, every obtainable **COG** organism pair was presented by one averaged protein size. It was done by averaging all paralog protein sizes to keep only one representative size per protein per organism. Typical distribution of lengths of proteins in the **COG** is rather homogeneous. Therefore, even the presence of “ill-defined” **COG**s cannot negatively affect the results of our natural statistical procedure.

8.2.3 Information Bottleneck Algorithm

According to the fundamental Information Bottleneck (IB) approach [250], the natural statistical measure of the information that variable X holds about variable Y is the mutual information $I(X;Y)$ of the random variables X and Y:

$$I(X;Y) = \underset{\Sigma}{\sum} x \in X, y \in Y = \underset{\Sigma}{\sum} x \in X, y \in Y,$$

where $p(x,y)$ is the joint distribution of X and Y; $p(x)$ and $p(y)$ are the marginal distributions of X and Y, rewspectively; $p(y|x)$ is the conditional distribution of Y given$X$. It is easy to show that $I(X;Y)$ is a symmetric and non-negative function, which equals zero if and only if the variables are independent. This parameter assesses the average number of bits needed to express the information that Xhas about $Y$and *vice versa*. Given the joint distribution $p(X;Y)$, the IB method seeks for a compact representation of X, which keeps as much information as possible about variable Y. Typically, $X$is the variable to be compressed, while $Y$is the variable to be predicted. The idea of the trade-off between two types of information terms was formulated [250] using the basic notions of the well-known Shannon’s information theory. According to this idea, the goal is keeping only those features of Xwhich

[1] An *outlier* is an item of a set that lies far from most other items in a set of data.

are most useful for predicting Y. Optimum representations are constructed by means of a supplementary variable Twhich stands for soft partitions of Xvalues, so that $I(T;X)$ is minimized when $I(T;\, Y)$ reaches its maximum. The compressed representation $T$of X(which, in our case, corresponds to clustering of X) is defined by $p(T|X)$. Hence, the quality of the clusters is calculated using the information which is covered by Y, namely, by $I(T;\, Y)/I(X;\, Y)$. The distribution of $T$is determined given Xalone because $T$is a compressed representation of this variable:

$$p(T|X,Y) = p(T|X),$$

or

$$p(X,Y,T) = p(X,Y)p(T|X).$$

The above expressions can be reformulated using the equivalent form of the IB

Markovian relation:

$$T \leftrightarrow X \leftrightarrow Y.$$

The IB optimization task can be reduced to the minimization of the IB functional:

$$L[p(T|X)] = I(T;X) - \beta I(T;Y),$$

where β is a positive Lagrange multiplier. The minimization is performed over all the $p(T|X)$ distributions. The minimization problem has an exact analytical formal solution without any assumption regarding the joint distribution$p(X,Y)$. This solution is provided by means of the following three distributions :

$$\begin{bmatrix} p(t|x) = \frac{p(t)}{Z(\beta,x)} \exp(-\beta D_{KL}(p(y|x)||p(y|t))) \\ p(y|t) = \frac{1}{p(t)} \sum\limits_{x} p(t|x)p(x)p(y|x) \\ p(t) = \sum\limits_{x} p(t|x)p(x) \end{bmatrix},$$

where

- $p(t)$ is the prior cluster probability;
- $p(t|x)$ represents the membership probabilities;
- $p(y|t)$ denotes the distribution of the relevant variable;
- $Z(\beta,\, x)$ is a normalization factor;
- $\beta \quad \geq 0$ is the Lagrangian multiplier parameter, which establishes the tradeoff

between compression and accuracy and the partition.

In the context of genome clustering considered here, the similarity measure of two genomes is defined as the similarity between the conditional distributions of the representative lengths of their genes. The latter distributions have the following form:

$$p(y|x) = \frac{n(y|x)}{\sum\limits_{y_i \in Y} n(y_i|x)},$$

where $n(y|x)$ is the average length of the genes corresponding to COG y in genome x. We also make use of the uniform prior distribution of the genomes, $p(x) = 1\|X\|$.

It would be natural to suppose that genomes with similar conditional length distributions belong to the same cluster. This supposition leads to the hierarchical structure of genome clusters, which is based on the similarity of their conditional distributions. The idea of cluster hierarchy of the set items based on the similarity of their conditional distributions was first introduced in [198], where the sets $X$and Ywere associated with a set of documents and with a suitable collection of words, respectively. Such approach is referred to as *distributional clustering.* Obviously, this method requires introducing the definition of the distance between distributions which would reflect the desired measure of similarity. The Information Bottleneck principle determines the distortion measure between the points by means of the known Kullback-Leibler divergence between the conditional distributions $p(y|x)$ and $p(y|t)$:

$$D_{KL}(p(y|x)||p(y|t)) = \sum_y p(y|x) \log \frac{p(y|t)}{p(y|x)}.$$

The membership probabilities, $p(t|x)$, are generally “soft”, i.e., each element can be assigned to each cluster with some (normalized) probability.

Bolshoy and Volkovich used the agglomerative Information Bottleneck algorithm proposed in [237]. Let $p(x, y)$ be the mutual distribution of the genomes and the average length of the genes of the COGs. In this case, the merging criterion is based on

$$D_{JS}(x, t) = (p(x) + p(t)) * JS(p(y|x), p(y, t)),$$

where $JS(p, q)$ is the famous Jensen-Shannon divergence defined as

$$JS(p(y|x), p(y, t)) = \pi_1 D_{KL}(p(y|x)||\overline{p}) + \pi_2 D_{KL}(p(y|t)||\overline{p}),$$

where

$$\pi_1 = \frac{p(y|x)}{p(y|x) + p(y|t)}, \ \pi_2 = \frac{p(y|t)}{p(y|x) + p(y|t)}$$

and

$$\overline{p} = \pi_1 p(y|x) + \pi_2 p(y|t).$$

The Jensen-Shannon divergence is non-negative and equals zero if and only if both its arguments are identical. The following two procedures construct a partition with exactly K clusters.

```
**Input:** The joint probability distribution, $p(x, y)$ - the mutual distribution of the genomes $X$ and the average length of the genes of the COGs $Y$.
**Output:** Partition of the genomes into $K$ clusters, for each $K : 1 \leq K \leq |X|$.
**Initialization:**
• $\bar{X} = X$;
• for each $i, j = 1, ..., |X|$, $i < j$ calculate $d_{ij} = D_{JS}D(p(y|x_i)p(y|x_j)))$
**Procedure:**
• for $m = |X| - 1$ to 1
- Find the indices $i, j$; for which $d_{ij}$ is minimized;
- Merge $\{x_i, x_j\}$ to $\bar{x}$;
- Update $\bar{X} = (X - \{x_i, x_j\}) \cup \bar{x}$
- Update $d_{ij}$ with respect to $\bar{X}$
• End for
```

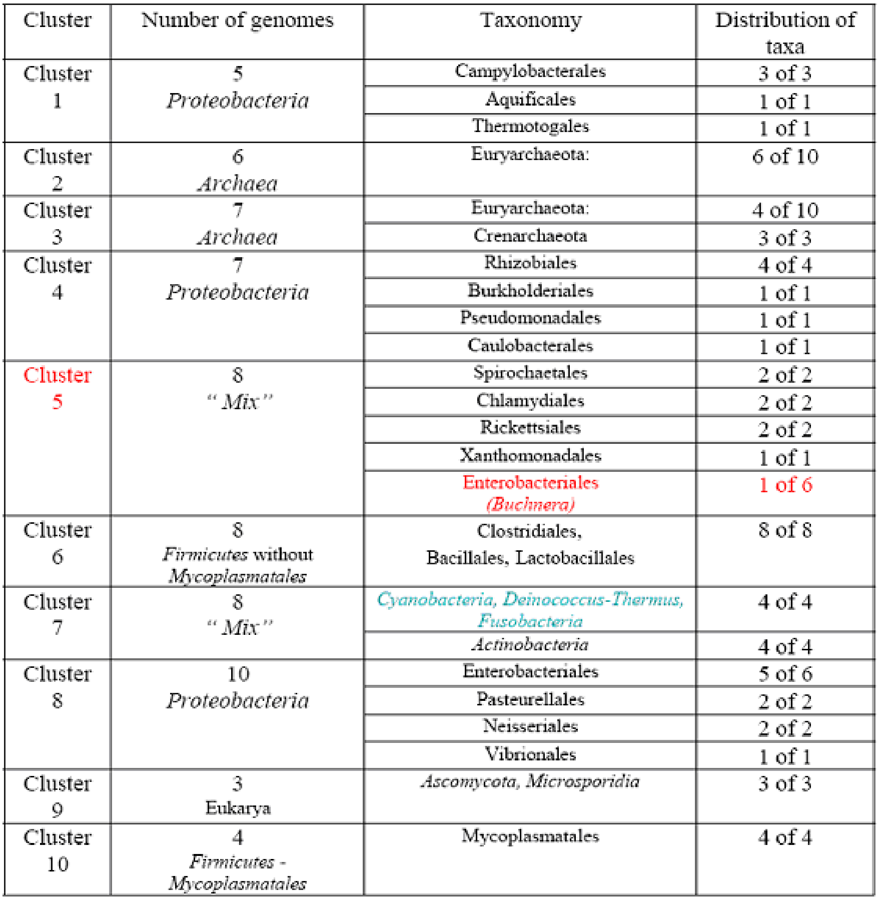
```
**Input:** The joint probability distribution, $p(x, y)$, and the average length of the genes of the COGs $Y$.
**Output:** Partition of the genomes into $K$ clusters, for each $K : 1 \leq K \leq |X|$.
**Initialization:**
• $\bar{X} = X$;
• for each $i, j = 1, ..., |X|$, $i < j$ calculate $d_{ij} = D_{JS}D(p(y|x_i)p(y|x_j)))$
**Procedure:**
• for $m = |X| - 1$ to 1
- Find the indices $i, j$; for which $d_{ij}$ is minimized;
- Merge $\{x_i, x_j\}$ to $\bar{x}$;
- Update $\bar{X} = (X - \{x_i, x_j\}) \cup \bar{x}$
- Update $d_{ij}$ with respect to $\bar{X}$
• End for
```

8.3 Clustering Obtained Using the IB Methods and Its Biological Significance

In the previous section the IB methods were described. We applied these clustering procedures using distances based on the following two properties:

- presence-absence of genomes in **COG**s;
- distribution of protein lengths in **COG**s.

The answers to two questions were sought: whether usage of IB improves former clustering results obtained in the framework of presence-absence of orthologous genes; and, whether the addition of important evolutionary information hidden in protein length distribution makes clustering more conventional, more traditional? Distributions of the genomes among the clusters are shown in Fig.8.1. First of all, let us compare the results with traditional taxonomy. Both applications result in a clear separation of the two major prokaryotic domains, Bacteria and Archaea (Figs. 8.1, 8.2). A comparison



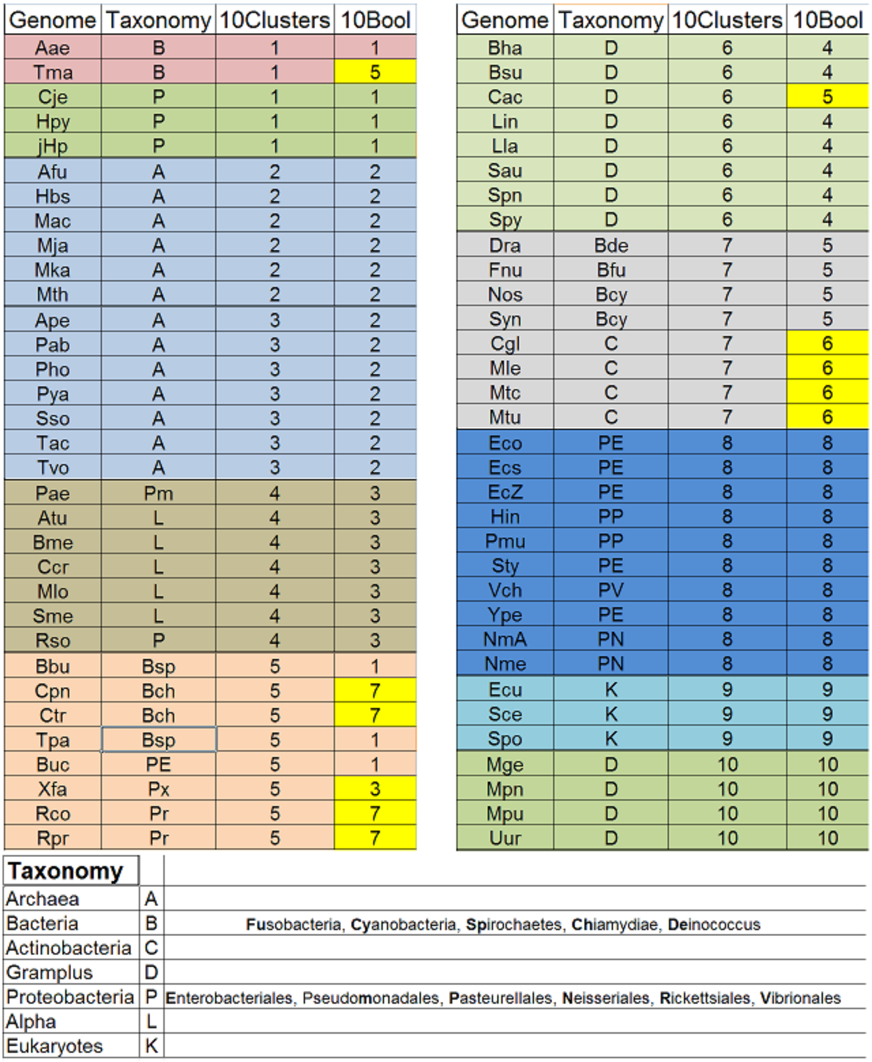
Cluster	Number of genomes	Taxonomy	Distribution of taxa
Cluster 1	5 *Proteobacteria*	Campylobacterales	3 of 3
		Aquificales	1 of 1
		Thermotogales	1 of 1
Cluster 2	6 *Archaea*	Euryarchaeota:	6 of 10
Cluster 3	7 *Archaea*	Euryarchaeota:	4 of 10
		Crenarchaeota	3 of 3
Cluster 4	7 *Proteobacteria*	Rhizobiales	4 of 4
		Burkholderiales	1 of 1
		Pseudomonadales	1 of 1
		Caulobacterales	1 of 1
Cluster 5	8 *" Mix"*	Spirochaetales	2 of 2
		Chlamydiales	2 of 2
		Rickettsiales	2 of 2
		Xanthomonadales	1 of 1
		Enterobacteriales *(Buchnera)*	1 of 6
Cluster 6	8 *Firmicutes* without *Mycoplasmatales*	Clostridiales, Bacillales, Lactobacillales	8 of 8
Cluster 7	8 *" Mix"*	*Cyanobacteria, Deinococcus-Thermus, Fusobacteria*	4 of 4
		Actinobacteria	4 of 4
Cluster 8	10 *Proteobacteria*	Enterobacteriales	5 of 6
		Pasteurellales	2 of 2
		Neisseriales	2 of 2
		Vibrionales	1 of 1
Cluster 9	3 Eukarya	*Ascomycota, Microsporidia*	3 of 3
Cluster 10	4 *Firmicutes - Mycoplasmatales*	Mycoplasmatales	4 of 4

Fig. 8.1 Clustering to 10 groups based on gene lengths

of the species in the list presented in Table 8.1,8.2 and Fig.8.1 provides the following results:

- All 13 Archaeal genomes are placed together in the union of clusters 2 and 3.
- Crenarchaeota – all three genomes appear together in one cluster.
- Eukaria – all three genomes appear together in one cluster.
- Cyanobacteria – both genomes appear together in one cluster.

Generally, taxonomically closely related species, such as four mycoplasmas (*M. genitalium, M. pneumoniae, M. pulmonis,* and *U. urealiticum*), two spirochetes (*B. burgdorferi* and *T. pallidum*), and two close organisms *H. pylori* and *C. jejuni,* are found in the common clusters.



Genome	Taxonomy	10Clusters	10Bool
Aae	B	1	1
Tma	B	1	5
Cje	P	1	1
Hpy	P	1	1
jHp	P	1	1
Afu	A	2	2
Hbs	A	2	2
Mac	A	2	2
Mja	A	2	2
Mka	A	2	2
Mth	A	2	2
Ape	A	3	2
Pab	A	3	2
Pho	A	3	2
Pya	A	3	2
Sso	A	3	2
Tac	A	3	2
Tvo	A	3	2
Pae	Pm	4	3
Atu	L	4	3
Bme	L	4	3
Ccr	L	4	3
Mlo	L	4	3
Sme	L	4	3
Rso	P	4	3
Bbu	Bsp	5	1
Cpn	Bch	5	7
Ctr	Bch	5	7
Tpa	Bsp	5	1
Buc	PE	5	1
Xfa	Px	5	3
Rco	Pr	5	7
Rpr	Pr	5	7
Bha	D	6	4
Bsu	D	6	4
Cac	D	6	5
Lin	D	6	4
Lla	D	6	4
Sau	D	6	4
Spn	D	6	4
Spy	D	6	4
Dra	Bde	7	5
Fnu	Bfu	7	5
Nos	Bcy	7	5
Syn	Bcy	7	5
Cgl	C	7	6
Mle	C	7	6
Mtc	C	7	6
Mtu	C	7	6
Eco	PE	8	8
Ecs	PE	8	8
EcZ	PE	8	8
Hin	PP	8	8
Pmu	PP	8	8
Sty	PE	8	8
Vch	PV	8	8
Ype	PE	8	8
NmA	PN	8	8
Nme	PN	8	8
Ecu	K	9	9
Sce	K	9	9
Spo	K	9	9
Mge	D	10	10
Mpn	D	10	10
Mpu	D	10	10
Uur	D	10	10

Taxonomy		
Archaea	A	
Bacteria	B	**Fu**sobacteria, **Cy**anobacteria, **Sp**irochaetes, **Ch**iamydiae, **De**inococcus
Actinobacteria	C	
Gramplus	D	
Proteobacteria	P	**E**nterobacteriales, Pseudo**m**onadales, **P**asteurellales, **N**eisseriales, **R**ickettsiales, **V**ibrionales
Alpha	L	
Eukaryotes	K	

Fig. 8.2 10Bool column corresponds to the clustering based on presence/absence of genomes in COGs. 10Clusters correspond to the distribution of protein lengths in COGs clustering.

8.3.1 The Root of the Tree

The agglomerative information variant of the information bottleneck algorithm, as it follows from the name, sequentially divides the dataset in $2, 3 \ldots |X|$ groups, which means that the complete genome tree may be easily reconstructed. The root of the hidden genome tree and results of the partition to ten clusters are present in Fig. 8.1, and Figs. 8.2 and 8.3. In the whole

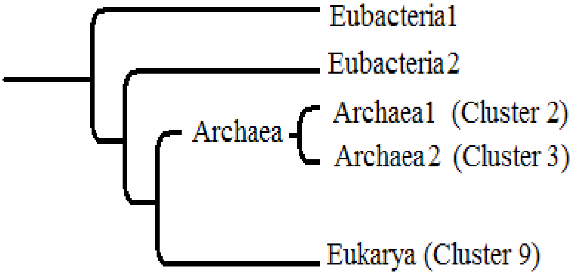


Fig. 8.3 The root of the genome tree of 66 unicellular genomes.

tree, clusters 1 and 10 (mycoplasmatales, campylobacterales, and bacterial hyperthermophiles) belong to one clade; clusters 5 and 6 form one big "mix" group, and clusters 4 and 8 belong to one unit as well. The application of the sequential clustering supports the accepted paradigm of the evolution of life expressed in the "Archaea tree" hypothesis (Chapter 2, compare Fig. 8.3 with Figs. 2.2 and 2.3).

8.3.2 What Does Clustering Based on the Presence-Absence of Genes Give Us

The data presented here in Fig. 8.2 in column "10Bool" are not identical to those reported by Wolf *et al.* (Fig. 3 in [287]). It appears that the partitions obtained using the IB method, which employs the presence-absence of genomes in COGs, better reflect phylogenetic relations than the results obtained in the methods of [287]. This may result from the fact that such partitions allow the avoidance of certain obstacles described in [287]. Therefore, the application of the IB method has already provided better results within the framework of gene-content clustering methods. However, usage of additional information related to gene attributes improves the results even more. In general, the differences between the columns "10Clusters" and "10Bool" (Fig. 8.2), which relate to the two different input datasets, are in favor of the method based on gene lengths.

Wolf *et al.* [287] consider the evolutionary affinity between Cyanobacteria (*Synechocystis*) and Actinomycetes (*Mycobacterium*) and also between two hyperthermophilic bacteria, *Aquifex* and *Thermotoga*, as plausible. According to our approach, which is not related to any of the five methods used in [287], Cyanobacteria (*Synechocystis* and *Nostoc*) and Actinomycetes (*Corynebacterium* and *Mycobacterium*) are located in the same cluster (cluster 7), while *Aquifex* and *Thermotoga,* together with three campylobacterales are all found in Cluster 1 (see Fig. 8.2). Thus, our results confirm those of Wolf *et al.* The position of Archaea in the genome tree (Fig. 8.3) is noteworthy. On the one hand, Fig. 8.3 shows separation of Bacteria and Archaea, Eukarya into two groups (which is in accordance with certain phylogenetic hypotheses); on the other hand, Archaea is split in a less traditional manner: (*Euryarchaeota: (Archaeoglobi, Halobacteria, Methanobacteria, Methanococci, and Methanomicrobia), (Euryarchaeota: (Thermoplasmata, Thermococci*, *Crenarchaeota*). To confirm previously obtained genome trees [287], *Buchnera aphidicola* does not appear together with taxonomically related *Escherichia coli* and *Salmonella typhimurium*, which is located in the “mix”-cluster 5.

8.3.3 Summarizing Conclusions

The ability to correctly cluster representatives of the major bacterial subdivisions and the absence of obviously wrong groupings confer credibility to non-trivial clades present in Table 8.2. In particular, convincing correspondence among earlier results of [222] and presented here groupings (for example, the *Spirochete + Chlamydia + Rickettsia* clade, non-trivial bacterial groupings *Aquifex* and *Thermotoga* together in one cluster, and *Cyanobacteria + Mycobacterium + Deinococcus* together in one cluster), seem to make the genome tree produced by the information bottleneck method a real representative of a species tree. It was also mentioned that this genome tree corresponds reasonably to the Archaea tree hypothesis. The results of the present study suggest that genome trees based on new clustering techniques and different types of whole-genome data may contribute to the further development of the field. Bolshoy and Volkovich in [250] and we here narrowed usage of genomic data to lengths of homologous proteins; however, there are several research groups that plan to expand the approach to data related with sequence similarity.

Appendix A
Clustering Methods

A.1 Clustering

A.1.1 Introduction

The objective of *cluster analysis*, or unsupervised classification, is dividing the data (objects, instances, items) into groups (clusters) in such a way that the items belonging to the same group are more similar to one another than to all the other items. Thus, clustering can be informally defined as *the procedure of systematizing objects into groups whose elements are similar in some way.* How can the quality of clustering be assessed? Obviously, the quality criterion should be dependent on the aim of the process and must be constructed in such a way that the result of the clustering would meet the requirements. Several goals could be relevant here. For example, it may be the determination of homogeneous group characteristics (data reduction) or the identification of "natural clusters" and the evaluation of their properties ("natural" data types), or the detection of abnormal items (outlier identification).

Generally speaking, most of the existing clustering methods can be categorized into three groups: partitioning, hierarchical, and density-based approaches. The advantage of partitioning methods is their ability to incorporate the knowledge about the cluster size by using certain templates and the elements' dissimilarity in the objective function. Such an algorithm inevitably produces clustering for any dataset, even if the data, have no cluster structure. This is due to the fact that, currently, there is no generally accepted way of testing the "null hypothesis" - the supposition about the absence of the cluster structure. The existing hierarchical clustering procedures yield a nested sequence of partitions and, as a rule, do not require the specification of the number of appropriate clusters. Instead, the partition is achieved by cutting the tree (dendrogram) at some level. Inner statistical tests (see the review in [118]) can hardly serve as a guide to determine the point for cutting the dendrogram. On the other hand, partitioning methods may produce a tighter cluster structure than hierarchical ones and are computationally faster for a larger number of variables in the case of a small number of clusters. Partitioning methods do not usually allow to obtain good results with non-globular clusters and the difference between various methods lies in the strategies of making a compromise in order to find suboptimal solutions. In fact, different methods could yield diverse results and, even with a specific method, the solutions are usually sensitive to initial conditions.

Except for the data itself, two essential input parameters are usually preset at the beginning of an iterative clustering procedure: the number of clusters and the initial partition. Usually, partitioning algorithms make various indirect assumptions about the dataset structure. For instance, the well known k-means clustering algorithm suggests that the considered set consists of a number of separate subsets of data points, spherically distributed around their average. One does not know whether these assumptions are true for the data, and the final partition produced by this algorithm largely depends on the initial partition of the data. Finding the ”correct” number of clusters in a dataset is an ill-posed task of cluster analysis [118], [94]. For example, this quantity can depend on the scale in which the data is measured (see, for example, [40]). Many approaches have been suggested to handle the problem. So far, none of them has been accepted as superior to others because cluster configurations are often very complex.

The following notations are used below:

- $\boldsymbol{X}$ is a finite subset of the Euclidian space $\boldsymbol{R}^n$ to be clustered. The elements of $\boldsymbol{X}$ are represented as $\mathbf{x} =(x_1, .., x_n)$;
- $N = |\boldsymbol{X}|$ is the size of the set $\boldsymbol{X}$;
- $\boldsymbol{R}_+ = [0, +\infty)$;
- $< \cdot, \cdot >$ is the inner product of two elements of $\boldsymbol{R}^n$;
- $tr(A)$ is the trace of matrix A;
- k is the number of clusters assessed;
- C_k is the set $\{1, ..., k\}$;
- Ψ_k is the set of all possible permutations of C_k;
- $\chi\,(A)$ is the indicator function of event A;
- For a clustering algorithm Cl, we suppose that the source data Z and the assumed number of clusters k are included in the input parameters. The output of the algorithm is the partition $\Pi_k(Z)$ of the set Z, which is represented by a labeling function $\alpha(\Pi_k) : Z \to C_k$;
- P_X is the distribution of a random variable X, while $P_{X,i}$ designates the probability of X being equal to i;
- ϕ is the density function of the standard normal distribution;
- Φ is the cumulative probability function of the standard normal distribution.

A.1.2 Dissimilarity Measures

An essential parameter of a clustering procedure is the *dissimilarity measure* between the data items. The success of the method depends on the measure choice, which is not always obvious, especially in multi-dimensional spaces. An appropriate way of introducing such a measure is using a metric function $d : \boldsymbol{X} \times \boldsymbol{X} \to \boldsymbol{R}^1$ which satisfies the following conditions:

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ (non-negativity);
2. $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$;
3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (symmetry);
4. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (triangle inequality)

for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \boldsymbol{X}$. Generally, in cluster analysis a metric is considered only as a framework for dissimilarities. In many applications not all of the above conditions are required. For instance, condition 2 may be replaced by: $d(\mathbf{x}, \mathbf{y}) = 0$ only if $\mathbf{x} = \mathbf{y}$ (a semi-defined metric). Some of the well-known examples of distance functions are:

- the Euclidian distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2};$$

- the Manhattan distance

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|;$$

- the Max distance

$$d(\mathbf{x}, \mathbf{y}) = \max_i |x_i - y_i|;$$

- the Minkowski distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}},$$

  where $p \geq 1$. Note that for $p = 2$, $p = 1$, and $p = \infty$ the Euclidian, the Manhattan and the Max distances are, respectively, obtained as partial cases.
- The correlation dissimilarity

$$d(\mathbf{x}, \mathbf{y}) = (1 - R(\mathbf{x}, \mathbf{y}))/2,$$

  where $R(x, y)$ is a correlation coefficient. In particular, $R(\mathbf{x}, \mathbf{y})$ can be the Pearson, the Spearman rank, or the Kendall rank correlation coefficient.

If the "items" to be clustered can be interpreted as probability distributions, probability metrics can be employed, for example

- the Kolmogorov - Smirnov (KS) metric

$$d(\mathbf{x}, \mathbf{y}) = \max_i |X_i - Y_i|,$$

where X_i and Y_i are two relative cumulative distributions, namely,

$$X_i = \sum_{j=1}^{i} \mathbf{x}_j, \; Y_i = \sum_{j=1}^{i} \mathbf{y}_j;$$

- relative entropy or the Kullback-Leibler (KL) divergence

$$d(\mathbf{x}, \mathbf{y}) = D_{KL}(\mathbf{x}||\mathbf{y}) = \sum_{i=1}^{n} x_i \log_2 \left(\frac{x_i}{y_i}\right). \tag{A.1}$$

  The usual convention is that $0 \log_2 \left(\frac{0}{y}\right) = 0$ for all real y, and $x \log_2 \left(\frac{x}{0}\right) = \infty$ for all real non-zero x. The relative entropy is not a metric since it is not symmetric and does not satisfy the triangle inequality.
- The Jensen-Shannon (JS) divergence with respect to the positive weights $\{\eta_1, \eta_2\}$

$$d(\mathbf{x}, \mathbf{y}) = D_{JS}(\mathbf{x}||\mathbf{y}) = \eta_1 D_{KL}(\mathbf{x}||\mathbf{z}) + \eta_2 D_{KL}(\mathbf{y}||\mathbf{z}), \tag{A.2}$$

  where $z = \eta_1 \mathbf{x} + \eta_2 \mathbf{y}$. The metric D_{JS} is symmetric, non-negative and upper-bounded. It equals zero if and only if $\mathbf{x} = \mathbf{y}$. However, it does not satisfy the triangle inequality.

A.1.3 Hierarchical Clustering

Hierarchical clustering procedures result in nested clusters, which may range from a single cluster coinciding with the set $\boldsymbol{X}$ to N clusters, each consisting of a single object. These procedures can be divided into *agglomerative* (*bottom-up*) or *divisive* (*top-down*) stratergies. Divisive (top-down) algorithms start from the whole set and successively separate the items into finer partitions. Agglomerative (bottom-up) methods, which are more often used, produce series of fusions of the data elements into groups. The traditional representation of the cluster hierarchy is a two-dimensional diagram tree (a dendrogram), which demonstrates the fusions or divisions that occur at each stage. This diagram has individual elements at one end and a single cluster, containing all the elements, at the other.

Given a distance matrix $N * N$, the agglomerative hierarchical clustering process defined by S.C. Johnson [122] can be descried as follows:

1. Let us consider each element to be a cluster. In this case, there exist N clusters and the intra-cluster distances coinside with the distances between the items.

2. The closest (in terms of distances) pair of clusters is determined and joined into a common cluster. As a result, the number of clusters decreases by one.
3. The distances between the new cluster and each of the other clusters are calculated and the new $(N-1)*(N-1)$ distance matrix is obtained. Note that the distances between clusters consisting of more than one item should be defined separately, the definition being another important parameter of the clustering process.
4. Steps 2 and 3 are repeated until all the items are merged into a single cluster.

An example of a dendrogram is given in Fig. A.1.

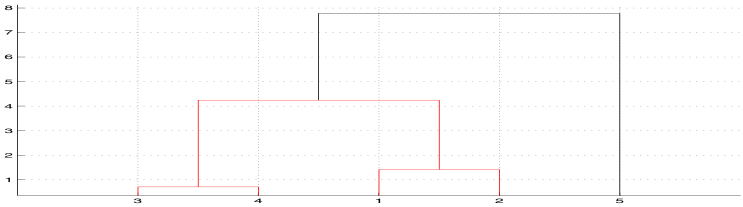


Fig. A.1 Dendrogram which illustrates the hierarchical clustering process for 5 items.

Below, a few ways of defining the distances between clusters consisting of more than one item are described. All the definitions are based on the initial $N * N$ distance matrix and are illustrated in Fig. A.2. The distance between item i, $i = 1, 2, 3$ of cluster 1 and item j, $j = 1, 2$ of cluster 2 is denoted by d_{ij}.

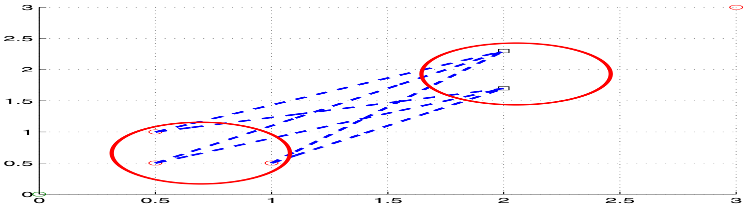


Fig. A.2 Different definitions of inter-cluster distances.

A.1.3.1 Single-Linkage Clustering

In this method, the distance between two clusters is defined as the minimal distance between any two items which belong to different clusters. For example (see Fig. A.2),

$$d(C_1, C_2) = d_{32} = \min\{d_{ij}; i = 1, 2, 3, j = 1, 2\}.$$

A.1.3.2 Complete Linkage Clustering

In this method, the distance between two clusters is defined as the greatest distance between any two items in different clusters. For example (see Fig. A.2),

$$d(C_1, C_2) = d_{21} = \max\{d_{ij}; i = 1, 2, 3, j = 1, 2\}.$$

A.1.3.3 Average Linkage Clustering

In this case, the distance between two clusters is defined as the average distance between any two items which do not belong to the same cluster. For example (see Fig. A.2),

$$d(C_1, C_2) = \overline{d} = \frac{1}{6}\sum_{i=1}^{3}\sum_{j=1}^{2} d_{ij}.$$

A.1.3.4 Ward's Linkage Clustering

The linkage function defining the distance between any two clusters is evaluated as the increase in the "error sum of squares" (ESS), caused by merging two clusters. In the Ward's method, two clusters are merged in such a way that the increase in ESS is minimized at every step. For a given set X, ESS is expressed as

$$ESS(X) = \sum_{\mathbf{x} \in \boldsymbol{X}} \|\mathbf{x} - \overline{\mathbf{x}}\|^2,$$

where $\overline{\mathbf{x}}$ is the average of the set elements. Formally, the Ward's distance between clusters C_1 and C_2 is calculated as

$$d(C_1, C_2) = ESS(C_1 \cup C_2) - ESS(C_1) - ESS(C_2).$$

Additional linkages are the sum of all intra-cluster variances and the V-linkage. They relate to the probability of the candidate clusters coming from the same distribution. Since these approaches are less common, we do not describe them here.

A.1.4 Partitional Clustering

In contrast to the hierarchical approach, partitional clustering methods involve a single partition of the data and do not produce dendrogram-like structures. Such methods are often based on solving a certain optimization problem. In particular, let us consider the partition

$$\boldsymbol{\mathit{\Pi}}_k = \{\pi_i,\ i = 1, ..., k\}$$

of the set $\boldsymbol{X}$, i.e.,

$$\boldsymbol{X} = \bigcup_{i=1}^{k} \pi_i \ \ and \ \ \pi_i \cap \pi_j = \emptyset \ if \ \ i \neq j.$$

The elements of the partition are referred to as clusters. For a real-valued function q, whose domain is the collection of all subsets of X, the quality of the partition is given by

$$Q\left(\mathbf{\Pi}\right) = \sum_{j=1}^{k} q(\pi_j). \tag{A.3}$$

A clustering problem is a particular case of a global optimization problem of determining the partition

$$\mathit{\Pi}^{(0)} = \{\pi_j^{(0)},\ j = 1, ..., k\}$$

which optimizes $Q(\mathit{\Pi})$. The function q can be built by means of a distance-like function $d(x, y)$ in the following way. Let us consider a predefined set of k centroids (medoids) $\boldsymbol{C}$=($\mathbf{c}_1, ..., \mathbf{c}_k$) as representatives of the clusters. The partition of $\mathbf{X}$ is constructed by determining

$$\pi_j = \{\mathbf{x} \in \boldsymbol{X} : d(\mathbf{c}_j, \mathbf{x}) \leq d(\mathbf{c}_i, \mathbf{x}),\ for\ i \neq j\},\ j = 1, ..., k.$$

(Ties are broken arbitrarily). On the other hand, for a given partition, the centroid set satisfies the condition

$$c(\pi_j) = \arg\min_{c}\{\sum_{x \in \pi_j} d(\mathbf{c}, \mathbf{x})\}.$$

Thus,

$$q(\pi_j) = \sum_{x \in \pi_j} d(c(\pi_j), \mathbf{x}),$$

which links the above-mentioned optimization problem to finding the appropriate set of centroids satisfying the condition

$$\boldsymbol{C} = \arg\min_{\mathbf{c} \in \boldsymbol{C}}\{\sum_{j=1}^{k} \sum_{x \in \pi_j} d(\mathbf{c}_j, \mathbf{x})\}. \tag{A.4}$$

A.1.4.1 *k*-Means Clustering

Let us suppose that $d(\cdot, \cdot)$ is the squared standard Euclidean distance function. In this case, the optimization problem (A.4) can be represented in the form:

$$\min_{\boldsymbol{C}} R(\boldsymbol{C}) = \sum_{j=1}^{k} \sum_{\mathbf{x}\in \boldsymbol{X}} \min_{c_j} \|\mathbf{x} - c_j\|^2 \tag{A.5}$$

An approximate solution of this optimization problem is obtained with the k-means algorithm, which includes the following steps [74]:

Input:
$\boldsymbol{X}$ - the set to be clustered;
k - the number of clusters;
$\mathbf{\Pi}_k^{(0)}$ - an initial (optional) partition.
Output: the partition $\mathbf{\Pi}_k$ of the set $\boldsymbol{X}$ into k clusters.

1. Initialization:
   Unless $\mathbf{\Pi}_k^{(0)}$ is not preset, an initial partition is constructed (items are usually randomly assigned to clusters).
2. Minimization: the mean value (centroid) of each cluster is calculated.
3. Classification: each element is assigned to the nearest centroid.
4. Steps 2 and 3 are repeated until the partition is stable, i.e., the centroids do not change any longer.

Proving the k-means clustering algorithm convergence is based on establishing the following two statements:

- Reassigning a point to a different group does not increase the error function.
- Updating the mean value of a group does not increase the error function.

However, the k-means algorithm often yields a partition consisting of the so-called *non-optimal stable clusters* . To overcome this difficulty, the incremental k-means algorithm can be used (see, e.g.,[68], [61], [59], [60], [145]). This algorithm gives better results in the case of relatively small clusters and it is often avoids local minima. Below, we briefly outline the main idea of the approach described in the papers cited above.

To illustrate the convergence of the k-means algorithm to *non-optimal stable clusters*, we consider the set $\boldsymbol{X} = \{0, 4, 7\}$. It is easy to see that an initial partition $\Pi_2^{(0)}$ represented by $\pi_1 = \{0, 4\}$, $\pi_2 = \{7\}$ in Fig. A.3 is not changed by the k-means procedure described above. However, the partition defined as $\pi_1 = \{0\}$, $\pi_2 = \{4, 7\}$ is superior to $\Pi_2^{(0)}$. Thus, no partition better than $\Pi_2^{(0)}$ has been detected by the algorithm in the case of the particular initial partition algorithm.

For a given partition $\boldsymbol{\Pi}_k$, an incremental version of the algorithm checks all the partitions $\widehat{\boldsymbol{\Pi}}_k(\mathbf{x})$ produced from $\boldsymbol{\Pi}_k$ by removing any point $\mathbf{x}$ from the cluster π_i and assigning it to the cluster π_j. The change of the objective function (A.3) is evaluated as

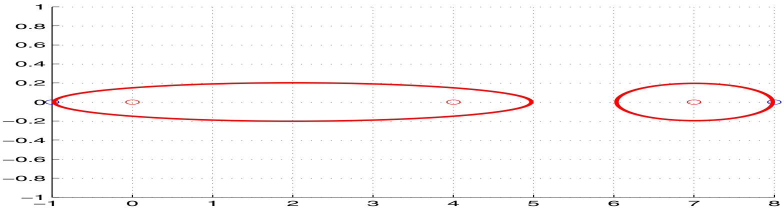


Fig. A.3 Initial Partition.

$$\Delta_{ij}\left(\boldsymbol{\Pi}_k, \mathbf{x}\right) = Q(\boldsymbol{\Pi}_k) - Q(\widehat{\boldsymbol{\Pi}}_k^{(ij)}(\mathbf{x})) = \tag{A.6}$$
$$\frac{|\pi_i|}{|\pi_i|-1}\left\|\mathbf{c}(|\pi_i|) - \mathbf{x}\right\|^2 - \frac{|\pi_j|}{|\pi_j|+1}\left\|\mathbf{c}(|\pi_j|) - \mathbf{x}\right\|^2.$$

To calculate the value of $\Delta_{ij}\left(\boldsymbol{\Pi}_k, \mathbf{x}\right)$, one needs to know the cluster sizes and the distances from each point to all the centroids. The latter distances are considered at each iteration of the k-means algorithm. This fact simplifies the application of a series of k-means iterations followed by an incremental iteration. Thus, the incremental step is performed when the k-means process has converged to a local optimum solution.

```
Incremental step algorithm:
Input:
$\boldsymbol{\Pi}_k$ - the initial partition;
$Dis-$ the set of the distances from each point of $\boldsymbol{X}$ to the centroids of $\boldsymbol{\Pi}_k$.
Output:
Refined partition $\widetilde{\boldsymbol{\Pi}}_k$
Algorithm:
for $i = 1$ to $k$
for $j = i + 1$ to $k$
  for $x \in \boldsymbol{X}$ do
    Calculate $\Delta_{ij}\left(\boldsymbol{\Pi}_k, \mathbf{x}\right)$ according to (A.6)
      If $\Delta_{ij}\left(\boldsymbol{\Pi}_k, \mathbf{x}\right) < 0$, then
          Move the point $\mathbf{x}$ from the cluster $\pi_i$ to the cluster $\pi_j$
          Substitute $\widehat{\boldsymbol{\Pi}}_k^{(ij)}$ for $\boldsymbol{\Pi}_k$
 end for
end for
end for
```

The meta algorithm can be described as follows:

Input:
$\boldsymbol{X}$ - the set to be clustered;
k - the number of clusters;
Tol - the tolerance.

```
Output:
Final refined partition $\widetilde{\widetilde{\boldsymbol{\Pi}}}_k$.
Algorithm:
Choose randomly $\widetilde{\boldsymbol{\Pi}}_k$
$Q(\Pi_k) = \inf$
Until $Q(\Pi_k) - Q(\widetilde{\boldsymbol{\Pi}}_k) > Tol$
  $[\Pi_k,\ Dis] = kmeans(\boldsymbol{X}, k, \widetilde{\boldsymbol{\Pi}}_k)$
  $\widetilde{\boldsymbol{\Pi}}_k = Incremental\ step(\Pi_k, Dis)$
end
$\widetilde{\widetilde{\boldsymbol{\Pi}}}_k = \widetilde{\boldsymbol{\Pi}}_k$
```

In the algorithm description, $kmeans(\boldsymbol{X}, k, \widetilde{\boldsymbol{\Pi}}_k)$ and $Incremental$ $step(\Pi_k, Dis)$ denote the applications of the k-means procedure and the incremental step described earlier, respectively. The version the k-means procedure provides the partition of the set $\boldsymbol{X}$ into k clusters on the basis of the initial partition $\widetilde{\boldsymbol{\Pi}}_k$. The incremental step receives, as its input parameters, the final partition Π_k, obtained by the k-means procedure, which employed the distance matrix Dis.

A.1.4.2 Expectation-Maximization Algorithm

The k-means approach can be considered as a simplification of the well-known Expectation-Maximization (EM) approach. The most widespread clustering version of the algorithm assumes the Gaussian Mixture Model (GMM) of data fitting (see, for example [16], [39], [76]):

$$f(x) = \sum_{j=1}^{k} p_j G(x|\mu_j, \Gamma_j), \tag{A.7}$$

where $G(x|\mu, \Gamma)$ is the Gaussian density with the mean value of μ and the covariance matrix Γ. Mixture models can be built in using the EM algorithm [58], which is a general iterative method, yielding maximum likelihood estimates when the data can be viewed as incomplete [72], [185], [186].

The model parameters are usually estimated on the basis of the data and may either vary for different clusters or be the same for all the clusters. The classification of several covariance models can be found in [76], [16]. The EM algorithm maximizes the following log likelihood function of parameters:

$$l = \sum_{\mathbf{x} \in \boldsymbol{X}} \log\left(\sum_{j=1}^{k} p_j G\left(x|\mu_j, \Gamma_j\right)\right).$$

In the EM clustering, the $N \times k$ matrix Z is introduced in such a way that, for each item $x_i \in \boldsymbol{X},$ $i = 1, ..., N,$ the coordinates of the row vector

$z_i = (z_{1i}, ..., z_{ki})$ are the probabilities that the item belongs to each of the k clusters under consideration.

Input:
$\boldsymbol{X}$ - the set to be clustered;
k - the number of clusters;
$\mathbf{\Pi}_k^{(0)}$– the initial partition (optional).
Output:
The partition $\mathbf{\Pi}_k$ of the set $\boldsymbol{X}$ into k clusters.
Algorithm:

1. **Initialization.** Initialize the mixture model parameters, thus creating the current model. The vectors z_i can be initialized by randomly assigning the value 1 to one element of each vector.
2. **M-STEP.** Recalculate model posterior parameters obtained by the E-STEP or by the initialization step, as follows:

   - The sample cluster sizes are calculated:
   $$\widehat{N}_j = \sum_{i=1}^{N} z_{ij}.$$
   - The sample cluster probabilities are calculated:
   $$\widehat{p}_j = \frac{\widehat{N}_j}{N}.$$
   - The sample-cluster mean values are calculated:
   $$\widehat{\mu}_j = \frac{\sum_{i=1}^{N} z_{ij} x_i}{\widehat{N}_j}.$$
   - The sample covariance matrix $\widehat{\Gamma}_j$ is calculated for each cluster.

3. **E-STEP.** Calculate the posterior probabilities from the Bayesian rule:
$$z_{ij} = \frac{\widehat{p}_j G\left(x_i|\widehat{\mu}_j, \widehat{\Gamma}_j\right)}{\sum_{j=1}^{k} \widehat{p}_j G\left(x_i|\widehat{\mu}_j, \widehat{\Gamma}_j\right)}.$$
4. **Convergence criteria testing.** Terminate if the current and new models are sufficiently close, else go to 2.

It should be noted that the classification EM or the CEM algorithm [39] already convert the values of z_{ij} to a discrete classification before the M-step. The standard k-means algorithm can be viewed as a version of the

CEM algorithm for the case of the uniform spherical Gaussian model with $\Gamma_j = \sigma^2 I,\ j = 1, ..., k$. Consequently, the clusters are spherically centered at the means $\mu_i,\ i = 1, .., k$ and the values of $p_j,\ j = 1, ..., k$ are implied to be equal.

A.1.4.3 $k-$Medoids or Partitioning around Medoids Algorithm

The Partitioning Around Medoids algorithm (PAM) [135] is a clustering procedure where the input represents the preset $N * N$ dissimilarity matrix. As an output, the algorithm generates a set of cluster centers or *medoids*, which, themselves, are the elements of the set being clustered. This is a beneficial property of PAM owing to the fact that the algorithm can be applied to any specified distance metric. In addition, the medoids are robust representations of the centroids. This is particularly important in the case when many elements cannot be precisely assigned to any cluster.

Taking into consideration (A.4), the corresponding objective function to be minimized may be represented in the form:

$$\min_{\boldsymbol{C} \in \boldsymbol{X}} R(\boldsymbol{C}) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in \boldsymbol{X}} \min_{c_j \in \boldsymbol{X}} d(\mathbf{c}_j, \mathbf{x}). \tag{A.8}$$

The PAM algorithm, which provides an approximate solution of the problem (A.8), consists of two phases. The first phase, **BUILD**, constructs the initial partition, while the second phase, **SWAP**, refines the partition.

Input:
Dis - the $N * N$ dissimilarity matrix of the items to be clustered;
k- the number of clusters;
$\mathbf{\Pi}_k^{(0)}-$ the initial partition (optional).
Output:
The partition $\mathbf{\Pi}_k$ of the set $\boldsymbol{X}$ into k clusters.
Algorithm:

```
1. Initialization: If $\mathbf{\Pi}_k^{(0)}$ is not given, then consecutively build the medoid
   set $\boldsymbol{C}$ which minmizes (A.8) (BUILD phase).
2. Until no change, do (SWAP phase)
3. Assign each element to the nearest medoid
4. for each $\mathbf{c} \in \boldsymbol{C}$ and for each $\mathbf{x} \in \boldsymbol{X} \setminus \boldsymbol{C}$

   a. compute the total cost, $S$, of swapping medoid $\mathbf{c}$ with $\mathbf{x}$
   b. if $S < 0$, then swap $\mathbf{c}$ with $\mathbf{x}$ to create the new set of medoids

5. end loop until
```

A.1.5 The Comparison of the Algorithms

A.1.5.1 k-Means Algorithm

- Relatively efficient: $O(kN)$ for each iteration;
- Non-robust with respect to noisy data and outliers;
- Cannot be used to separate non-convex clusters;
- Often converges to a local optimum;
- Requires predefining the number of clusters;
- Resulting clusters can be unbalanced or even empty (in the Forgy's version).

A.1.5.2 PAM Algorithm

- PAM is more robust than the k-means algorithm in the case of noisy data having outliers;
- PAM is efficient for small datasets, but does not scale well for large sets;
- Relatively inefficient: $O(k(N-k)^2)$ for each iteration;
- Requires predefining the number of clusters.

A.2 Information Clustering

A.2.1 Mixture Clustering Model

In this section, we discuss the Mixture Clustering model from the information-theory point of view [215], [240]. The main assumption of the model is that there is certain probability for each item in $\boldsymbol{X}$ to belong to each cluster. Here, the partiting solution is given by the set of probability distributions for the items which are associated in the same cluster. This association is termed *"fuzzy membership in clusters"*. The situation where each item belongs to one cluster is the hard clustering case. Determining the optimal association distribution is the goal of the probabilistic clustering approach. From the information theory point of view, clustering is the basic strategy for the data lost compression. According to this approach, the data is divided into groups, which are described, in the most efficient way, in terms of the bit rate, employing a representative of each group. In this section, we use the following notations:

For two discrete random variables, X and Y, taking the values $\{x_i\}$, $i = 1, 2, ...$ and $\{y_j\}$, $j = 1, 2, ...$, respectively, we use the *mutual information*

$$I(X,Y) = \sum_{i,j} p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i)p(y_j)} = \sum_{i,j} p(x_i, y_j) \log_2 \frac{p(y_i|x_i)}{p(y_j)},$$

where

- $p(x_i, y_j)$ is the joint probability function of X and Y;
- $p(x_i)$ and $p(y_j)$ are the marginal probability functions of X and Y;
- $p(y_i|x_i)$ is the conditional probability function of Y on X.

The clustering procedure is intended to compress the initial data by eliminating insignificant information. The relevant information is identified with the help of a distortion function, which usually measures the similarity of the data items. Let us suppose that, in each cluster, the distortion function is $d_j(\mathbf{x}, \mathbf{y})$, $j = 1, ..., k$. *Lossy compression* can be generated by assigning the data to the clusters so that the mutual information

$$I(\Pi, \boldsymbol{X}) = \sum_{x,j} P(\pi_j|\mathbf{x}) P(x) \log_2 \frac{P(\pi_j|\mathbf{x})}{P(\pi_j)} \tag{A.9}$$

is minimized. The minimization is constrained by fixing the expected distortion

$$\overline{d}(\Pi, \boldsymbol{X}) = \sum_{x,j} P(\pi_j|x) P(x) d_j(\mathbf{c}_j, \mathbf{x}),$$

where $\mathbf{c}_j$ is the representative(centroid) of the cluster π_j. In this case, the formal solution is obtained using the Boltzmann distribution

$$P(\pi_j|\mathbf{x}) = \frac{P(\pi_j)}{Z(\mathbf{x}, T)} \exp\left(-\frac{d_j(\mathbf{c}_j, \mathbf{x})}{T}\right),$$

where

$$Z(\mathbf{x}, T) = \sum_j P(\pi_j) \exp\left(-\frac{d_j(\mathbf{c}_j, \mathbf{x})}{T}\right)$$

is the normalization constant and T is the Lagrangian multiplier. In the hard-clustering case, the partition Π is defined by a set of associations

$$v(x, \pi_j) = \begin{cases} 1 \text{ if } x \in \pi_j \\ 0 \text{ otherwise} \end{cases}.$$

Thus, the underlying distribution of $\boldsymbol{X}$ is given by

$$\mu_{\boldsymbol{X}} = \sum_{j=1}^{k} p_j \mu_{\pi_j}, \tag{A.10}$$

where p_j, $j = 1, .., k$ are the cluster probabilities and μ_{π_j}, $j = 1, .., k$ are the inner cluster distributions. The marginal distribution of cluster centroids equals

$$P(\boldsymbol{C}) = \frac{e^{-\frac{F}{T}}}{\sum\limits_{Y} e^{-\frac{F}{T}}},$$

where

$$F = -T \sum_x \ln \left(\sum_j \exp \left(-\frac{d_j(\mathbf{c}_j, \mathbf{x})}{T} \right) \right)$$

is the so-called *"free energy"* of the partition. Evidently, the most probable values of the centroids minimize F. Therefore, the optimal estimates of the cluster parameters can be achieved by minimizing the free energy. An important example of this construction is the partition that corresponds to the distortions

$$d_j(x, \mathbf{c}_j) = (\mathbf{x} - \mathbf{c}_j)\Gamma_j^{-1}(\mathbf{x} - \mathbf{c}_j),$$

where $\mathbf{c}_j$ is the centroid of the cluster π_j and Γ_j is its covariance matrix. In this case, the expression for F can be rewritten as

$$F = -T \sum_x \ln \left(\sum_j \exp \left(-\frac{(\mathbf{x} - \mathbf{c}_j)\Gamma_j^{-1}(\mathbf{x} - \mathbf{c}_j)}{T} \right) \right).$$

From the equations

$$\frac{\partial F}{\partial \mathbf{c}_j} = 0, \ j = 1, ..., k,$$

at the fix matrices Γ_j, the final result can be obtained:

$$\mathbf{c}_j = \frac{\sum\limits_x \mathbf{x} P(\pi_j|\mathbf{x})}{\sum\limits_x P(\pi_j|\mathbf{x})}, \ j = 1, ..., k.$$

It should be pointed out that this result is similar to the maximum-likelihood estimation of normal mixture parameters obtained by means of the EM algorithm (see section A.1.4.2). The significant feature of the free energy optimization approach is that no prior information on the data distribution is required. The distributions are directly derived from the corresponding Bolzman and Gibbs distributions and the appropriate optimization task.

A.2.2 The Information Bottleneck Method

The Information Bottleneck method is a clustering technique which assumes that clusters have to reflect only the *relevant* information on the data. According to this approach, each element of $\boldsymbol{X}$ is identified with a distribution over its features and the partitions which are created on the basis of the empirical conditional distributions of the features

$$p(x_i|\mathbf{x}) = \frac{x_i}{\sum_{i=1}^n x_i}$$

are considered. The relevance of the information is established by the data features. Co-occurrence data may be organized according to this principle.

Some examples can be the occurrences of verbs and direct objects in sentences [250], words, documents [15], [106], [237], or tissues and gene expression patterns [251]. In the case of text clustering, $\boldsymbol{X}$ can be a set of documents and Y can be a set of words. Each element of $\boldsymbol{X}$ is represented by conditional distributions of words

$$p(y|x) = \frac{n(y|x)}{\sum\limits_{y_i \in Y} n(y_i|x)},$$

where $n(y|x)$ is the total occurrence of the word y in the document x. To finalize the model, a uniform initial distribution of the documents

$$p(\mathbf{x}) = \frac{1}{|X|}$$

is assumed. It would be natural to suppose that the documents which have close term conditional distributions belong to the same cluster. This supposition can lead to a cluster hierarchy structure based on the similarity of conditional word distributions. Generally speaking, any clustering method that employs such probability interpretation has to use a "probability metric" between distributions. The examples of such metrics are given in section A.1.2. The creation of a cluster hierarchy based on the similarity of conditional distributions was first introduced in [198].

Tishby, Pereira, and Bialek [250] suggested a method which avoids the arbitrary choice of a distortion or a distance measure. Given the joint distribution $p(X;Y)$, the method allows to obtain a compact representation of X, which keeps as much information as possible on the applicable variable Y. It is known that the mutual information between the random variables X and Y (see (A.9)) represents the natural statistical measure of the information that variable X holds about variable Y. The compressed representation T of X (which is, in our case, the clustering of X) is defined by $p(T|X)$. Hence, the quality of the clusters is calculated on the basis of the information which is covered by Y, namely, $I(T;Y)/I(X;Y)$.

The Information Bottleneck method allows to determine the distortion measure between the points by means of the well-known Kullback-Leibler divergence $D_{KL}(p(y|x)||p(y|t))$ between the conditional distributions $p(y|x)$ and $p(y|t)$. Generally, the membership probabilities, $p(t|x)$, are "soft", i.e., each element can be assigned to each cluster with some (normalized) probability. In the hard-clustering case, the agglomerative information bottleneck algorithm, proposed in [237], [236], uses the merging criterion based on the distortion:

$$DJS(x,t) = (p(x) + p(t)) * D_{JS}(p(y|x), p(y,t)),$$

where $D_{JS}(p,q)$ is the Jensen-Shannon divergence with identical weights.

A.3 Cluster Validation

One of the input parameters required by the iterative clustering algorithms is the assumed number of clusters in the considered dataset. The estimation of this number represents an ill-posed problem of crucial relevance in cluster analysis [118], [94]. For instance, the 'correct' number of clusters in a dataset can depend on the scale on which the data is measured [40].

Solution methods for this problem can be roughly divided into two groups:

- Methods based on the geometrical properties of clusters, such as within- and between-cluster dispersion.
- Methods based on the stability concept.

The methods that can be attributed to the first group were described by Calinski-Harabasz [35], Hartigan [100], Krzanowski-Lai [158], and Sugar-James [241], [93], [188], [94]; the Gap statistical method was proposed by Tibshirani, Walter and Hastie [248].

The stability of clusters is usually estimated from their variability under repeated application of a clustering algorithm to (random) samples from the same data source. Low variability in partitions is interpreted as high validity of the obtained result [44]. Thus, the number of clusters that corresponds to the maximal cluster stability can serve as an estimate for the "true" number of clusters.

Levine and Domany [168], Ben-Hur, Elisseeff, and Guyon [18], and Ben-Hur and Guyon [19] measured stability as the relative number of cases when a pair of elements occurs in the same cluster under repeated runs of the clustering algorithm. Bel Mufti, Bertrand, and Moubarki [190] determined the stability function based on the Loevinger's measure of isolation. The prediction-based resampling method (Clest) of Dudoit and Fridlyand [69], uses, actually, the external partition correlation index as a stability measure. Roth, Lange, Braun, and Buhmann [216] and Lange, Roth, M. Braun, and Buhmann [164] theoretically justify the solution of cluster validation problem by means of the stability concept. In the suggested model, the pairs of clustered samples are compared and the stability is defined as the relative rate of an element occurrences in the same cluster. Jain and Moreau [119] chose the dispersions of empirical distributions as the stability measure. Tibshirani and Walther [249] considered the cluster validation problem from the prediction-strength point of view.

A.3.1 Geometrical Criteria

Given the partition Π_k, $k \geq 2$, the total dispersion matrix (the total scatter matrix) is given by

$$T_k = \sum_{j=1}^{k} \sum_{z \in \pi_j} (\mathbf{z} - \overline{\mu})(\mathbf{z} - \overline{\mu})^t, \tag{A.11}$$

where $\overline{\mu}$ is the arithmetic mean of the set X. Matrices B_k and W_k of between- and within-k-cluster square sums are defined as

$$B_k = \sum_{j=1}^{k} |\pi_j| \, (\overline{\mu}_j - \overline{\mu})(\overline{\mu}_j - \overline{\mu})^t, \; W_k = \sum_{j=1}^{k} \sum_{z \in \pi_j} (\mathbf{z} - \overline{\mu}_j)(\mathbf{z} - \overline{\mu}_j)^t, \quad \text{(A.12)}$$

where $\overline{\mu}_j$ is the arithmetic mean of π_j, $j = 1, ..., k$. Note that $T_k = W_k + B_k$ [179].

The following inner indices are frequently used to estimate the number of clusters in a dataset.

1. The Calinski and Harabasz index [35] is defined as

$$CH_k = \frac{tr(B_k)/(k-1)}{tr\,(W_k)\,/(N-k)},$$

where N is the size of the dataset under consideration. The estimated number of clusters is given by k, which corresponds to the maximum of CH_k.

2. The Krzanowski and Lai index [158] is defined as

$$diff_k = (k-1)^{2/n} tr(W_{k-1}) - k^{2/n} tr(W_k),$$

where n is the dimension of the dataset and

$$KL_k = |diff_k|/|diff_{k+1}|.$$

The estimated number of clusters corresponds to the maximal value of the index KL_k.

3. The Hartigan index [100] is defined as

$$h_k = \left(\frac{tr(W_k)}{tr(W_{k+1})} - 1 \right) (N - k - 1).$$

The estimated number of clusters is the smallest $k \geq 1$ such that $h_k \leq 10$, where N, again, is the size of the dataset under study.

4. Sugar and James [241] proposed an informational theoretic approach for finding the number of clusters in a dataset. According to their method, the differences of transformed distortions are calculated. Namely,

$$J_k = \left(tr(W_k)^{-t} - tr(W_{k-1})^{-t} \right),$$

where t is the transformation power. The estimated number of clusters corresponds to the maximal value of the index J_k.

5. In the case of the Gap index [248], the values of $tr\,(W_k)$ are calculated for each $k \geq 1$. Reference datasets are generated under the null distribution assumption (in [248], the reference dataset number, B=10). Each

of the datasets is subjected to the clustering procedure and the values of $tr\left(W_k^1\right), ..., tr\left(W_k^B\right)$ are evaluated. The estimated gap statistics are:

$$gap_k = \frac{1}{B}\sum_b \log\left(tr\left(W_k^b\right)\right) - \log\left(tr\left(W_k\right)\right).$$

Let sd_k be the standard deviation of $\log\left(tr\left(W_k^b\right)\right)$, $1 \le b \le B$, and

$$\hat{sd}_k = sd_k\sqrt{1+\frac{1}{B}}.$$

The estimated number of clusters is the smallest $k \ge 1$ such that

$$gap_k \ge gap_{k^*} - \hat{sd}_{k^*},$$

where $k^* = argmax_{k \ge 1}(gap_k)$. Two approaches are considered for constructing the region of support for the distribution (for details, see [248]).

A.3.2 Stability-Based Criteria

The determination of the possible number of clusters by means of resampling procedures has been considered by many authors (see, e.g., several recent contributions [168], [69], [81], and [216].) These approaches are based on the *cluster stability* concept . We describe here the two most widespread algorithms of this kind.

A.3.2.1 External Indexes

Below, we consider several external criteria used in cluster stability approaches. The calculation of these scores is based on the so-called *cross-tabulation* or *contingency tables.* The membership of elements in two partitions Π_r and Π_c of the same dataset is compared. Let N_{ij} denote the number of items which are the members of cluster i of Π_r and of cluster j of Π_c $(i = 1, ..., r,\ \ j = 1, ..., c)$. Note that the relationship between the two partitions can be considered in the framework of the measures of nominal data associations. The most well-known of such measures is the Cramer correlation coefficient, which is defined as follows. Let us introduce

$$N_i^{(r)} = \sum_{j=1}^{c} N_{ij},\ i = 1, ..., r,\ N_j^{(c)} = \sum_{i=1}^{r} N_{ij},\ j = 1, ..., c. \tag{A.13}$$

Obviously,

$$N = \sum_{i=1}^{r} N_i^{(r)} = \sum_{j=1}^{c} N_j^{(c)},$$

and the chi-square statistic equals

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(N_{ij} - e_{ij})^2}{e_{ij}}, \; e_{ij} = \frac{N_i^{(r)} * N_j^{(c)}}{N}. \tag{A.14}$$

The Cramer coefficient is

$$V = \sqrt{\frac{\chi^2}{N * \min(r-1, c-1)}}. \tag{A.15}$$

Several known external indexes employ the statistic

$$Z = \sum_{j=1}^{c} \sum_{i=1}^{r} N_{ij}^2. \tag{A.16}$$

Rand [210] introduced the index (R)

$$R = 1 + \left(\frac{Z - 0.5 * \left(\sum_{j=1}^{c} \left(N_j^{(c)} \right)^2 + \sum_{i=1}^{r} \left(N_i^{(r)} \right)^2 \right)}{\binom{N}{2}} \right);$$

Jain and Dubes [118] considered the index

$$JD = \frac{(Z - N)}{\left(\sum_{j=1}^{c} \left(N_j^{(c)} \right)^2 + \sum_{i=1}^{r} \left(N_i^{(r)} \right)^2 - Z - N \right)},$$

while Fowlkes and Mallows [75] proposed the following expression (FM):

$$FM = \frac{(Z - N)}{2 \sqrt{\sum_{j=1}^{c} \binom{N_j^{(c)}}{2} \sum_{i=1}^{r} \binom{N_i^{(r)}}{2}}}.$$

It is easy to see that two indexes, R and FM, are linear functions of Z; consequently, either index is a linear function of the other one.

An external index is often standardized in such a way that its expected value is 0 if the partitions are random and 1 when they match perfectly. The general formula for the standartization of an index is:

$$Ind' = \frac{(Ind - E(ind))}{(Ind_{\max} - E(ind))},$$

where $Ind_{\max}$ denotes the maximal value of the index Ind.

The most commonly used null model assumes that the contingency table is created from the generalized hyper-geometric distribution and that the two partitions are mutually independent. In this case, the adjusted index has to be zero. The index values close to zero mean that nothing can be predicted from each partition about the other.

The cluster stability problem has also been considered [18], [19] from the point of view of the areas of concentration of an external index distribution which is built on samples drawn from the dataset.

Next, we will consider the Clest method [69], which also uses an external index as a stability magnitude.

A.3.2.2 Clest Algorithm

In this method, the true number of clusters is assessed through sequential splitting the source dataset $\boldsymbol{X}$ into two non-overlapping subsets of the same size. These subsets, $\boldsymbol{L}_b$ and $\boldsymbol{T}_b$, are referred to as a *learning set* and a *test set* of the current iteration b, respectively. For each tested number of clusters k, a partition of the learning set is provided. This partition is used to predict the clustering of the test set, which is simultaneously divided into clusters by the direct application of the clustering procedure. The two partitions of the test set are evaluated using one of the external indices described above. For each k, the indices are compared to their expected values, calculated within a suitable null distribution in the absence of cluster structure. The true number of clusters corresponds to the largest significant "evidence" against the null hypothesis. A version of this method was proposed earlier by Breckenridge [30].

The algorithm can be described in the following way:

For each number of clusters k, $2 \leq k \leq k^*$, perform steps 1-4.

1. Repeat the following B times:

   a. Randomly split the original learning set into two non-overlapping sets, a learning set $\boldsymbol{L}_b$ and a test set $\boldsymbol{T}_b$.
   b. Apply the clustering procedure to the learning set $\boldsymbol{L}_b$ to obtain the partition $\pi\left(\boldsymbol{L}_b\right)$.
   c. Construct a classifier $C(\boldsymbol{L}_b)$, using $\pi\left(\boldsymbol{L}_b\right)$.
   d. Apply the classifier $C(\boldsymbol{L}_b)$ to the test set $\boldsymbol{T}_b$.
   e. Apply the clustering procedure to the test set $\boldsymbol{T}_b$ to obtain the partition $\pi\left(\boldsymbol{T}_b\right)$.
   f. Calculate an external index $s_{k,b}$ to compare the two sets of labels for $\boldsymbol{T}_b$.

2. Let

$$t_k = median(s_{k,1}, ..., s_{k,B})$$

   be the observed median value of the external index for the k-cluster partition of the data.

3. Produce B_0 datasets under an appropriate null hypothesis. For each dataset, repeat the procedure described in steps 1 and 2 to obtain B_0 statistics $t_{k,1}, ..., t_{k,Bo}$.
4. Consider the average of the B_0 statistics:

$$t_k^{(0)} = \frac{1}{B_0} \sum_{b=1}^{B_0} t_k$$

and denote by p_k the proportion of $t_{k,b}, 1 \leq b \leq B_0$ which are at least as large as the observed statistic t_k, i.e., the p-value for t_k. Let $d_k = t_k - t_k^{(0)}$ denote the difference between the observed similarity statistic and its estimated expected value under the null hypothesis. Introduce the set K as

$$K = \{2 \leq k \leq k^* : p_k \leq pmax, d_k \geq dmin\},$$

where $pmax$ and $dmin$ are predefined parameters. If this set is empty, no cluster structure is detected. Otherwise, the number of clusters, k, corresponds to the largest significant difference statistic d_k:

$$k = \arg\max_{k \in K} d_k.$$

The authors used the PAM algorithm, described in section A.1.4.3, the naive Bayes classificator, the FM index (see section A.3.2.1), $B = B_0 = 20$, and $pmax = dmin = 0.05$.

A.3.2.3 Stability-Based Validation of Clustering Solutions

In what follows, we present the algorithm for the internal index calculation described in [216].

1. Split the dataset of size $2t$ into two sets of equal size, $\boldsymbol{X}_1^t$ and $\boldsymbol{X}_2^t$.
2. Apply the algorithm to the first dataset. The result is the mapping α_1 of each item in $\boldsymbol{X}_1^t$ onto one of the k clusters.
3. Apply the algorithm to the second dataset $\boldsymbol{X}_2^t$. The result is the mapping α_2 of each item in $\boldsymbol{X}_2^t$ onto one of the k clusters. Use α_2 to predict the cluster membership of all the items contained in the first set.
4. Now, the set $\boldsymbol{X}_1^t$ has two different labelings. Find the correct permutation of the labels using the well-known Hungarian method for minimum weighted perfect bipartite matching [159]. The costs for identifying labels i and j are the number of misclassifications with respect to the labels α_1, which are assumed to be correct.
5. Normalize with respect to random stability.
6. Reiterate the whole procedure from step 1 to step 5, average over the assignment costs, and calculate the expected (in-)stability value.
7. Reiterate the whole procedure for each k to be tested.

The application of the Hungarian method is required since the cluster labels can be permuted. The match between the labels can be generated by solving the problem

$$\tau_k(\alpha_1, \alpha_2) = \min_{\psi} \frac{1}{t} \sum_{j=1}^{t} \chi\{\alpha_1(x_j) \neq \psi(\alpha_2(x_j))\}, \tag{A.17}$$

where $\psi \in \Psi_k$, Ψ_k is the set of all possible permutations of the set $\boldsymbol{C}_k$ (see above). The Hungarian method has a computational complexity of $O(k^3)$. Such a technique was applied in [216], [164] and also in the cluster-containing area [252].

Normalization with respect to the random stability means:

$$S_k = \frac{mean(\tau_k(\alpha_1, \alpha_2))}{mean(\tau_k(\rho_1, \rho_2))}, \tag{A.18}$$

where ρ_1, ρ_2 are random predictors which assign labels in a uniformly random way. The estimated number of clusters is given by k, which enables us to obtain the maximum of S_k.

The authors point out that splitting a dataset into two disjoint subsets is recommended because the size of individual sets should be large. However, this approach can be, formally applied for any sample size.

A.3.3 Probability Metric Approach

It has been noted earlier that the stability-based methods generate a variety of partitions for the same number of clusters. In addition, the majority of the known iterative clustering algorithms involve random initializations of the suitable optimization processes. Consequently, the partitions obtained using these algorithms can vary. From the statistical standpoint, these partitions can be considered as estimates of the sought-for "true partition". Thus, the partitions are viewed (see ......) as instances of a unique random variable such that the steadiest one of them is associated with the true number of clusters. This stability is measured by means of probability distances within the observed realization.

A.3.3.1 Probability Metric

In this section, we describe several facts from the probability metric theory, which can be found in [209], [294]. In general, probability metrics are introduced on the space of real-valued random variables, Λ, defined on the probability space (Ω, B, P). The functional $dis : \Lambda \to R_+$ is called a *probability metric* if it has the following additional properties:

1. Identity: $dis(X,Y)=0 \Longleftrightarrow P(X=Y)=1$;
   (Semi-Identity: $P(X=Y)=1 \Rightarrow dis(X,Y)=0$);
2. Symmetry: $dis(X,Y)=dis(Y,X)$;
3. Triangular inequality: $dis(X,Z) \leqslant dis(X,Y)+dis(Y,Z)$ for all random variables X, Y, and Z.

If a probability metric identifies a distribution (i.e., $dis(X,Y)=0 \Longleftrightarrow P_X=P_Y$)), the metric is called *simple*, otherwise the metric is called *compound*. Simple and compound metrics differ in that simple metrics, unlike compound ones, equal zero for two independent identically distributed random variables. Moreover, if *dis* is a compound metric, $dis\left(X_1,\, X_1^{'}\right)=0$, and $X_1,\, X_1^{'}$ are independent realizations of X, then X is almost certain to be a constan. On the other hand, a compound distance can be used as a measure of uncertainty. In particular, $dis\left(X_1,\, X_1^{'}\right)=d(X)$ is called a *concentration measure index*, derived from the compound distance *dis* [209]. The stability of a random variable can be estimated by the average value of this index.

Obviously, the requirements for a probability metric are similar to those for dissimilarity measures (see section (A.1.2)). Consequently, the distances which can be considered here may not satisfy the triangular inequality. Moreover, this feature will not be required in what follows.

Examples of simple probability metrics which, indeed, measure the distances between appropriate marginal distributions can be provided by the distances discussed in section (A.1.2). Many distances applicable to two-sample tests are of simple metrics. For instance, the Kolmogorov-Smirnov test is based on the max-distance between the probability functions in the one-dimensional case. The Friedman-Rafsky test [80], the nearest-neighbour test [104], the energy test [290], and the R-distance, proposed recently by Klebanov [143], can be mentioned in the context of the multivariate case.

A famous example of a compound metric is given by L^p-metrics which are similar to the Minkowski distance. For every $p \geqslant 0$, the L^p-metric is defined by

$$dis_p(X,Y)=E(|X-Y|^p)^{min(1,1/p)},$$

where

- $dis_\infty(X,Y)=\inf\{c>0 : P\left(|X-Y|>c\right)=0\}$;
- $dis_0(X,Y)=E(\chi(X \neq Y))$ is the indicator metric.

A.3.3.2 Algorithm

Let us consider a cluster stability criterion based on probability metrics for the case $\Omega = C_k$. Realizations of an appropriate random variable are simulated by rerunning a clustering algorithm on the same collection of items, X. The correspondence between two different labels is provided by the solution of the optimization problem (A.17). The clustering algorithm can be presented in the following way:

```
Choose the parameters:
    • $\boldsymbol{X}$ - the set to be clustered;
    • $N_s$ - the number of samples;
    • $M$ - the sample size;
    • $k^*$ - the maximal number of tested clusters;
    • $CL = \{Cl_i, i = 1, ..., 2N_s\}$ - the set of clustering algorithms.
for $k = 2$ to $k^*$
    for $n = 1$ to $N_s$
        $S^{(n)} = sample(\boldsymbol{X}, M)$
        $\Pi_k^{(2n-1)}(S_{(n)}) = Cl_{2n-1}(S_{(n)}, k),\ \Pi_k^{(2n)}(S_{(n)}) = Cl_{2n}(S_{(n)}, k)$
        for $\left(z \in S^{(n)}\right)$
            $X_n(\boldsymbol{x}) = \alpha(\Pi_k^{(2n-1)}, \boldsymbol{x}),\ Y_n(\boldsymbol{x}) = \alpha(\Pi_k^{(2n)}, \boldsymbol{x})$
        end for
        Permute $Y_n = \psi^*(Y_n)$ according to (A.17) so that labels of $Y_n$ match those of $X_n$
        $\widehat{DIS}_k^{(n)} = \frac{1}{M} \sum_{\boldsymbol{x} \in S^{(n)}} dis(X_n(\boldsymbol{x}), Y_n(\boldsymbol{x}))$
    end for
    Normalize $\{\widehat{DIS}_k^{(n)}\}$ with respect to the random stability
    $m(k) = mean(\widehat{DIS_k^{(n)}})$
end for
The estimate for the “true” number of clusters is the $k$ which minimizes $m(k)$.
```

A.4 Feature Selection

In the process of data partitioning, we are interested in the recovery of any cluster structure that arises in subspaces, especially, for the sake of facilitating visualization, in two- or three- dimensional subspaces. Clustering techniques usually employ information from all the features of elements which define their similarities as well as the dimensionality of the data space. However, using the full-dimentional space does not often prove to be the optimal clustering method, mainly in the case when the cluster structure is enclosed in a subspace. In this case, the ”residual” data may be very noisy and thus complicate the analysis of the cluster structure. Moreover, many of the features can be interdependent and, therefore, a new set of independent variables has to be found. Thus in many cases the redundant information must be excluded in order to obtain a more suitable representation of the data.

Other reasons for considering low-dimensional data projections are the so-called *curse of dimensionality* and the *empty space* phenomena. Generally speaking, the curse of dimensionality consists in the exponential growth of the full dimentionality with the number of the sample-size variables [17]. This growth is especially manifested in the case of estimating a function of several variables to the given degree of accuracy. The related empty space phenomenon usually shows as inherent sparsity of high-dimensional spaces [229]. Apparently, the above two difficulties may result in the appearence of an "empty data projection" or in the presence of different item features in orthogonal subspaces.

A.4.1 Principal Component Analysis

Principal Component Analysis (PCA) is a standard common approach to reducing the dimensionality of a dataset. The aim of PCA is to determine linear combinations of the element features that are operational in the description of their variation. Let us suppose that the data is represented by the $n \times N$ matrix $\boldsymbol{X}$ with a zero empirical mean value. Without loss of generality, the empirical mean value can be eliminated from the data set. Construct the $n \times n$ covariance matrix characterizing the data scatter in the form:

$$\Gamma = \frac{1}{N}\left(\boldsymbol{X} * \boldsymbol{X}^T\right).$$

Next, the eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_n$ and the corresponding normed eigenvectors $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n$ of Γ are computed. he only eigenvectors that are left are consequent to those which correspond to the m largest eigenvalues and represent each vector by:

$$\mathbf{x} = \sum_{i=1}^{m} b_i \mathbf{u}_i. \tag{A.19}$$

The value of m is usually chosen by taking into consideration only the eigenvalues greater than 1 or by using the criterion

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > Threshold(e.g., 0.8 or 0.9).$$

The PCA procedure can be formally described in terms of the spectral decomposition of the covariance matrix :

$$\Sigma = U \Lambda U^T,$$

where U is an orthogonal matrix having the eigenvectors $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n$ as its columns, and Λ is a diagonal matrix with the diagonal items $\lambda_1, \lambda_2, ..., \lambda_n$. The transform (A.19), in a general case, has the form:

$$\mathbf{b} = U^T\left(\mathbf{x} - \overline{\mathbf{x}}\right).$$

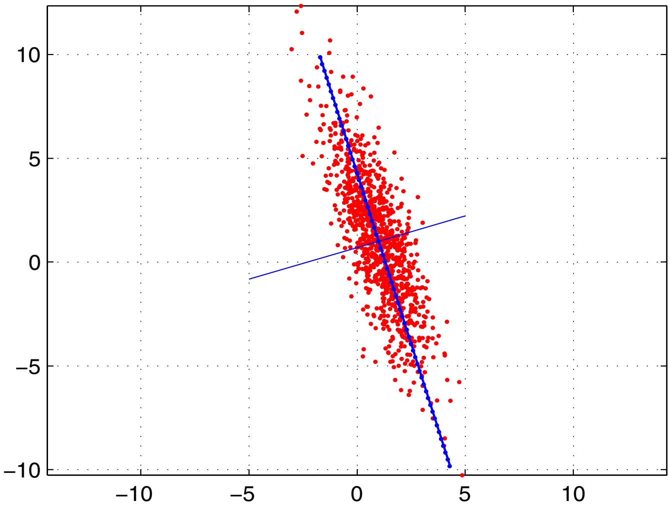


Fig. A.4 Normal scattered data together with the first two principal components.

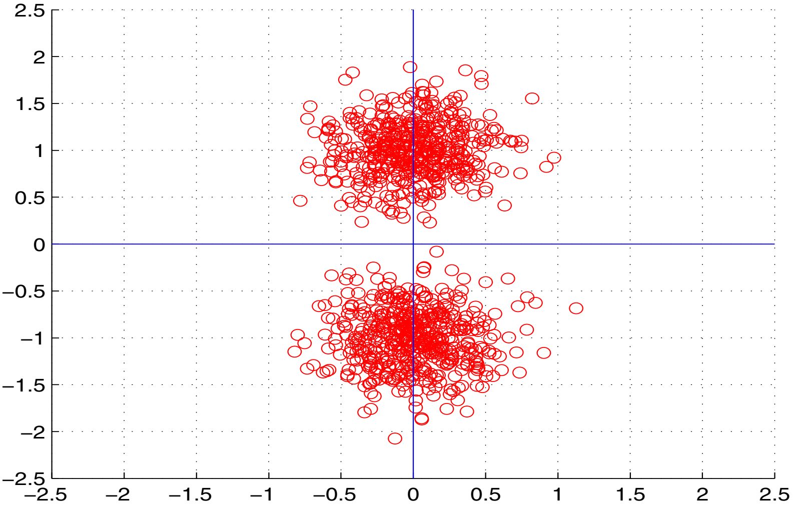


Fig. A.5 Scatter plot of bivariate data.

A few eigenvectors with the greatest eigenvalues are referred to as *principal components* , the first greatest eigenvalue corresponding to the first principal component, the second greatest eigenvalue corresponding to the second principal component , and so on.

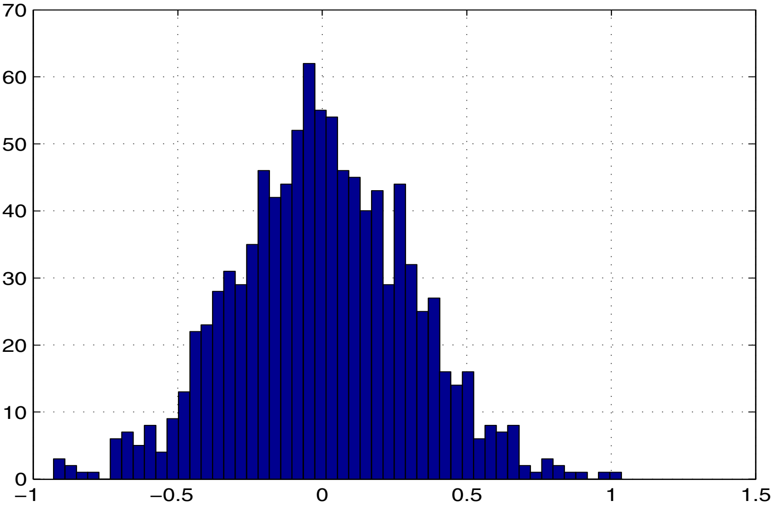


Fig. A.6 Histogram of the first principal component.

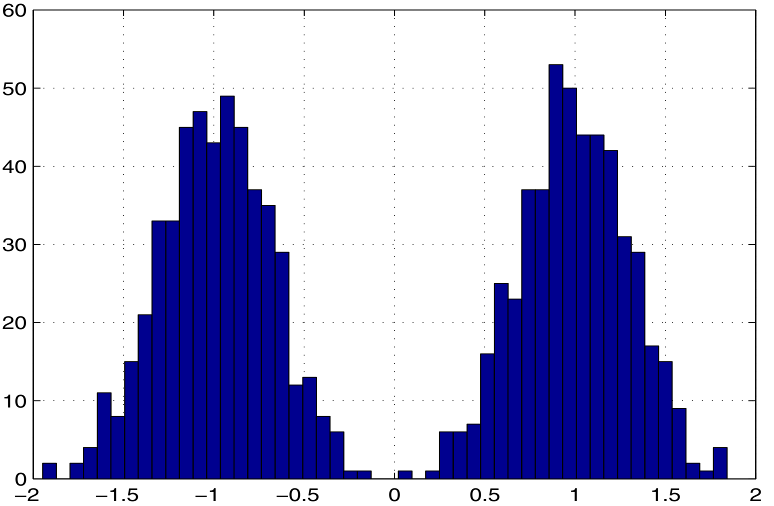


Fig. A.7 Histogram of the second principal component.

An example of normal scattered data, together with the first and the second principal components, is presented in Fig. A.4.

In many situations, the principal components account for a significant part of the information which is contained in the source set. However, some relevant information may be dropped out in this case. For instance, the information contained in a small set of components found by PCA can be useless for clustering tasks. The bivariate data presented in Fig. A.5 provide a famous example of such situation.

Histograms of the the first and the second principal components are shown in Figs. A.6 and A.7, respectively.

From the data presented in Fig. A.5, it can be seen that the first principal component coincides with the direction of the greatest data spread, while the second component is orthogonal to the first one (which is an inherent feature of proncipal components). The histogram of the first component (Fig. A.6) appears to be uni-modal and, therefore, provides no information about the cluster structure. The bi-modal histogram of the second component (Fig. A.7) reveals the existence of two clusters, which is really the case. Thus, the above example shows that, restricting the analysis to the first component, we lose information about the real cluster structure.

A.4.2 Projection Pursuit Techniques

Projection pursuit techniques are intended to discover those data projections that can best reveal the data structure. Such projections can be used for the detection of the most demonstrative representations of the data clustering structure. Projection pursuit is built by optimizing the predefined function which is called a *projection pursuit index*. It apears that the idea of a projection pursuit index was first introduced in [157] and [79]. The projection on a suitable low-dimensional subspace often makes it possible to overcome the curse-of-dimensionality problem. PCA can be viewed as a particular projection pursuit route where the index is the total data variance. An arbitrarily chosen projection of a high-dimensional dataset on a low-dimensional space can bias the sample so that it becomes comparable to a sample drawn from a normal distribution. It was demonstrated [62] that, in the high-dimensional case, distributions of linear projections are approximately normal under certain weak assumptions. According to the well-known Cramer-Wold principle, a multidimensional distribution having only normal one-dimensional projections is also normal. However, in a general case, a one-dimensional projection can prove to be an informative characteristic of the dataset. For example, a projection density distribution may exhibit peaks, each one corresponding to a cluster. Therefore, in unsupervised classification tasks, indices based on the so-called *"departure from normality"* are preferred. Many indexes have been introduced on the basis of various approaches. Below, we present only several most widespread methods. A review of dimensionality reduction techniques can be found, for instance, in [37].

A.4.3 L^2 Distance-Based Indexes

Many pursuit indexes are based on the L^2 distance between the projection density distribution and the appropriate normal distribution. The most well-known is the Friedman's Pursuit Index [78]. Let $\boldsymbol{X}$ be centered and sphered data. For a given direction $a \in \boldsymbol{R}^n$, consider $\boldsymbol{Y} =< a, \boldsymbol{X} >$ and introduce a new random variable

$$\mathbf{R} = 2\Phi(\boldsymbol{Y}) - 1,$$

where Φ is the cumulative standard normal function. This variable is mapped onto the interval $[-1, 1]$. If $\boldsymbol{Y}$ is the standard normal distribution, $\mathbf{R}$ is uniformly distributed on $[-1, 1]$ and *vice versa.* The density of $\mathbf{R}$ is given by:

$$p_{\boldsymbol{R}}(x) = \frac{\frac{1}{2} p_Y(\Phi^{-1}(\frac{x+1}{2}))}{\varphi(\Phi^{-1}(\frac{x+1}{2}))}, \tag{A.20}$$

where $p_{\boldsymbol{Y}}$ is the density of $\boldsymbol{Y}$; φ is the standard normal density. The direction corresponding to the density $p_R(x)$ which differs most from $\frac{1}{2}$ in the L^2 norm is derived by maximizing the functional

$$l(a) = \int\limits_{-1}^{1} \left(p_r(x) - \frac{1}{2} \right)^2 dx = \int\limits_{-1}^{1} p_r^2(x) dx - \frac{1}{2}.$$

The function $p_R(x)$ can be approximated by a series of the Legendre polynomials $L_k(x)$:

$$p_r(x) = \sum_{i=0}^{\infty} c_i L_i(x),$$

where

$$c_i = \frac{2i+1}{2} \int\limits_{-1}^{1} L_i(x) p_r(x) dx.$$

Substituting the approximations into the above expression for the index, we obtain

$$l(a) = \sum_{i=0}^{\infty} \frac{2i+1}{2} E(L_i(x))^2 - \frac{1}{2},$$

where E designates the expectation. This value can be estimated using the empirical expectation

$$\hat{E}(L_i(x)) = \frac{1}{N} \sum_{t=1}^{N} L_i(2\Phi(< a, \mathbf{x}_t >) - 1).$$

Since N is the size of the dataset under consideration, the required calculation can be easily performed using the recursive relationships:

$$L_0(x) = 1,\ L_1(x) = x,$$
$$L_i(x) = \frac{2i-1}{i} x L_{i-1}(x) - \frac{i-1}{i} L_{i-2}(x).$$

The series expansion in the expression for $l(a)$ should be truncated to avoid overfilling since the high-order estimated coefficients are unstable even for large samples. Typically, no more than eight terms are usually considered in the expansion. On the other hand, if the sample size is sufficiently large, we can start from a small number of terms and subsequently improve the local maxima by increasing the number of terms.

It was shown [53] that the Friedman index can be rewritten in the form:

$$l(a) = \int_{-\infty}^{\infty} \frac{(f(x) - \varphi(x))^2}{2\phi(x)} dx,$$

where $f(x)$ is the normalized projection density. Hence, this index is a special case of a general set of indexes based on the orthogonal polynomials considered in [53]. For instance, the Hall index [99] is given by

$$l_H(a) = \int_{-\infty}^{\infty} (f(x) - \varphi(x))^2 \, dx,$$

while the natural Hermit index has the form

$$l_C(a) = \int_{-\infty}^{\infty} (f(x) - \varphi(x))^2 \, \phi(x) dx.$$

The corresponding calculations can be performed using the orthogonal expansion of $f(x)$ on the basis of Hermit polynomials.

A.4.4 Entropy-Based Indexes

Another technique to measure the deviation from normality employs *differential entropy* [112], [123]. The differential entropy H of a random vector with the density f is defined as

$$H(f) = -\int f(x) \log(f(x)) dx.$$

The negentropy, which can be viewed as the natural information-theoretic one-unit contrast function, is defined as

$$J(f) = H(p_g) - H(f),$$

where p_g is the Gaussian density with the mean value and the covariance equal to those of f. Several entropy properties related to the pursuit index concept were discussed in [123]. In particular, Gibb's second theorem states that the multivariate Gaussian density maximizes the entropy with respect to f over all distributions with the same mean value and the same covariance. For any non-normal distribution, the consequent entropy is precisely smaller. So, negentropy is positive for non-normal distributions and equals zero if the distribution is Gaussian. Actually, the calculation of entropy is a complicated task.

Estimates of entropy tend to use polynomial expansions of the density or the Gram-Charlier-Edgworth approximations. For instance, negentropy for standardized random values can be evaluated by means of higher-order cumulants

$$J \simeq -\frac{1}{12}(k_3^2 + \frac{1}{4}k_4^2)$$

where k_i, $i = 3, 4$ are the i-th order cumulants. The approximations of negentropy suggested in [116] can be represented, in the simplest case, as

$$J \simeq c\,(E(V(x)) - E(V(g))),$$

where V is any non-quadratic function, c is an arbitrary constant, and g is a standardized Gaussian variable. For $V(x) = x^4$, the kurtosis modulus is obtained. The asymptotic variance and robustness of these estimations can be improved by a suitable choice of V. The following options for V are recommended:

$$V_1 = \log(\cosh(a_1 x)), \qquad V_2 = \exp(-\frac{a_2 x^2}{2}).$$

It was found empirically that the values $1 \leq a_1 \leq 2$, and $a_2 = 1$ provide especially good approximations.

A.4.5 BCM Functions

It was mentioned in the previous section that computationally attractive projection indices based on polynomial moments cannot be applied directly since they are extremely sensitive to the deviation from normality in the tails of the distributions. Friedman tried to overcome this problem by using a nonlinear projection of the data onto the interval $[-1, 1]$ (see above). On the other hand, the Friedman index is not sensitive to multi-modality in the projected distribution in the case of significant differences between the pick sizes. This insensitivity arises from the L^2 norm approximation.

An approach for exploring the projection multi-modality was considered in the framework of the synaptic modification neuron theory of Bienenstock, Cooper, and Munro (BCM). It yields synaptic modification equations that maximize the projection index $l(a)$ as a function of the direction a. In this context, $l(a)$, which measures the deviation from the Gaussian distribution, is called a *cost function*. Synaptic modification equations are evaluated using gradient ascent with respect to the weights (see, e.g., [20]).

Intrator and Cooper [117] introduced the following cost function to assess the deviation from the Gaussian distribution in the multi-modality:

$$l(a) = \frac{1}{3}E(< a, \boldsymbol{X} >^3) - \frac{1}{4}E^2(< a, \boldsymbol{X} >^2).$$

It was shown [20] that the cost function

$$l(a) = E\left[< a, \boldsymbol{X} >^2 (1 - \frac{1}{2} < a, \boldsymbol{X} >^2)\right]$$

can be regarded as the cost function for PCA. Other cost functions based on skewness and kurtosis of the projection can also be discussed.

Appendix B
Sequence Complexity

B.1 Motivation: Finding Zones of Low Complexity

One of the fundamental characteristics of any text is its complexity. Since the early days of bioinformatics, different measures of sequence complexity applicable to genetic texts have been proposed ([235, 258, 150, 151, 221, 222, 34, 261, 220]). Sequence-complexity-based methods may be used both for overall characterization of long genomic sequences and for their comparison. The repetitiveness of sequences in a genome may be visualized and studied by constructing the map of sequence complexity along the genome ([261]). In this way, it is possible to detect all zones of lower and higher complexity in long genomic sequences. In ([261]), the authors presented plots of complexity distribution along the *H. influenzae* genome using window sizes of 40, 100, and 2000 bases. A small-size sliding window was used for the identification of relatively short repeats, while a big window may cover long dispersed and degenerate repeats and reveal their role in decreasing the complexity. A sequence complexity map of a whole genome gives an overall view of its organization.

B.2 Compositional Complexity

The term *compositional complexity* was introduced by Konopka ([150, 146, 147, 149]). The numerical value of compositional complexity of a sequence depends solely on the size of the alphabet and on the frequencies of the occurrences of certain elements (monomers, dimmers, or N-grams over the chosen alphabet) in the sequence ([148]). One of the applications of compositional complexity to sequence analysis is finding functionally relevant properties through studying large collections of functionally equivalent sequence fragments ([149]). The significance of the results should be assessed relative to a background level expected only on the basis of chance. Usually, the results of a run on real data are compared to numerous runs on random data. In many applications, it is important to construct random data so that the local dinucleotide bias is preserved. Using the methods of shuffling biological sequences ([126]), one can obtain numerous realizations of randomized sets with the dinucleotide composition identical to that of the set under investigation. Therefore, to assess the difference between the compositional complexities of the given set and the one related to random data, shuffled sequences should be compared. Since multiple-fold shuffling may prove to be a computationally time-consuming task, an entropy-based algorithm for comparing compositional complexities of the given set and of random data was proposed (Bolshoy:2008). *Shannon uncertainty* (entropy) is a function of -$\log_2 q$, where, in our case, q is the frequency of a certain oligonucleotide of length six (6-tuple, hexamer) in a given text. Sequence complexity originated from the Shannon entropy and has at least three underlying elements: 1. The number of symbols in the alphabet. (Four potential nucleotide alphabetic symbols could occupy each mononucleotide position in a DNA sequence.) 2. The assumption that each symbol in any position has equal probabilistic availability. (Generally speaking, this assumption is incorrect for genetic sequences, but it works well enough as a zero hypothesis.) 3. Sequence length. (For the purpose of sequence comparison, all the above-mentioned measures of sequence complexity may be applied only to sequences of nearly identical sizes.) The *Shannon information* (entropy) has become a standard measure for the order state of DNA sequences ([235, 150, 151, 222, 147, 149, 226]). *Shannon information* has the form

$$I = \log_2 \lambda + \sum_{i=1}^{\lambda} p_i \log_2 p_i,$$

where λ is the size of the alphabet, p_i is the probability to find symbol A_i in an arbitrary position. For example, if $\lambda = 4$ (which is the case for the DNA alphabet) and the length of words is 6 (hexamers), we obtain:

$$I_6 = 12 + \sum_{i=1}^{4096} q_i \log_2 q_i,$$

where i is the index of the word A_i, q_i is the probability to find the word A_i in an arbitrary position, $\lambda^m = 4^6 = 4096, \log_2(4096) = 12$. This kind of sequence complexity measure is called *compositional complexity.* Within the framework of the compositional complexity approach, sequence order and complexity are at the opposite ends of the scale. The most complex sequence appears to be the most disorganized one, i.e., it has the minimum number of recognizable repetitions.

B.3 Waterloo Complexity

Ming Li at the University of Waterloo, Canada ([169]) proposed to use *Kolmogorov sequence complexity* as a similarity measure. $K(X|Y)$ is the *conditional Kolmogorov complexity* (or *algorithmic entropy*) of x given y. $K(x)$ is defined as the size of the smallest Turing machine (the length of the shortest algorithm), which generates x as an output on the input of y. For formal definitions, theorems, and discussions, see ([170]). The authors point out that the Kolmogorov complexity, viewed as the ultimate lower bound of all measures of information, can be just approximated, but not calculated in the general case. Li *et al.* ([169]) applied a measure of compressibility as an approximation of complexity. They used their own program GenCompress, which was, certainly, a rather voluntary choice. To compare two sequences, one has to calculate complexity values of both sequences, of the juxtaposition of both sequences, and the conditional complexity as well. In this method, sequence similarity is measured as a relative decrease of complexity or conditional complexity $K(X|Y)$ ([170]). The approach seems to be very promising, however (as it was also mentioned also in their pioneer work ([169]), an efficient procedure to compute a suitable estimation of the conditional complexity $K(X|Y)$ has yet to be developed.

B.4 Linguistic Complexity

Linguistic complexity (LC) (the term first coined by Trifonov [258]) is a measure based on counting all different overlapping words in the text. This method has the following advantages: (1) it is conceptually simple, (2) the calculations can be performed quickly, and (3) it has been successfully used in computational linguistics. The essence of the approach is a comparison of the actual number of different substrings to their maximal possible number in a given string. Originally, the approach was used by Popov *et al.* ([206]) to compare biological sequences to natural language texts and by Bolshoy *et al.* [24] to demonstrate that a weak pattern can be significantly enhanced by subset selection according to the sequence complexity criterion. In [83]. The authors showed that the parallel analysis of sequence the authors showed that the parallel analysis of sequence complexity and DNA curvature might provide important information about the sequence-structure-function relationships in prokaryotic genomes. In Troyanskaya *et al.* [261], LC was defined as the ratio of the actual number of substrings present in the string of interest to the maximum possible number of substrings in a string of the same length over the same alphabet. The actual number of different substrings in a string was calculated using the suffix trees, while the maximum vocabulary over the words of lengths in the range ($1m$) was calculated according to the following formula (where l is the alphabet size and k is the word length):

$$\sum_{k=1}^{m} min(l^k, m - k + 1).$$

The algorithm, based on implicit suffix trees constructed with the Ukkonens algorithm ([263]) in order to count the number of substrings in the string, provides an effective way to reveal variations in linguistic complexity of genomic texts. One useful application of this algorithm proved to be searching for lower-complexity zones. These regions are dispersed along the genome and manifest themselves by atypical pattern LC variation. The simplest, yet a common case of a low-complexity zone in prokaryotes, is a region of *simple sequence repeats* (SSR), which usually consists of homopolymeric tracts, predominantly of poly(A) or poly(T), and of more rare multimeric repeats of longer length. Functionally, SSRs participate in regulating gene expression at various levels ([270, 271, 269]). The examination of the patterns of sequence complexity around the flanks of coding sequences ([261]) revealed potential regulatory sites. It was found that the complexity profiles of GC-rich genomes (*D. radiodurans, M. tuberculosis*, and *T. pallidum*) differ substantially from those of AT-rich genomes. Frequent location of low-complexity zones in a close proximity to the boundaries of coding sequences is one of the major features of AT prokaryotic genomes. The construction of linguistic complexity profiles proved to be a useful instrument for detecting various genomic signals. For example, it was found that in AT prokaryotes, homopolymeric A- and T-tracts downstream from many genes are characteristic features of transcription terminators. Some of these tracts are traces of known intrinsic terminators; however, the reduction of linguistic complexity in the downstream non-coding region is also indicative of the presence of additional, earlier undetected transcription termination signals in this region.

Appendix C
DNA Curvature

C.1 DNA Curvature and Gene Regulation in Prokaryotes

The double-stranded DNA structure, which predominates in cells and is observed in many viruses, was introduced and discussed in Chapter 1. It is well known now that the double helix (Fig. C.1) can adopt different conformations, i.e. the parameters of the helix may vary within certain limits. It has been demonstrated experimentally that the twist angle - the angle of the base-pair rotation around the axis of the double helix - is not sequence-independent, as originally assumed in the Watson-Crick model (Chapter 1, Section 2.1). Actually, the twist angle of B-DNA may range between 30 and 40 degrees and, consequently, the number of base pairs per one helical turn may also vary depending on the environmental conditions and on the sequence of nucleotides.

Many of these alternative DNA structures were shown to have biological significance, e.g., they are recognized by proteins that regulate DNA replication and transcription (see Chapter 1, Section 4.2.2) [256, 259]. Consider, for example, the control of transcription initiation in prokaryotes, which is one of the major points of gene regulation. It has been shown that recognition of specific DNA sequences by proteins plays a pivotal role in transcription initiation. In Chapter 1, Section 3.2.1, the reader can find the description of the transcription initiation region (promoter), which includes two main fragments recognized by the enzyme DNA-dependent RNA polymerase (see Fig. 1.6). One fragment is located at about 10 bp and the other one at about 35 bp upstream from the transcription initiation site. It was shown that proteins recognize these specific DNA sequences not only through direct contacts between bases and amino acids, but also indirectly via DNA specific sequence-dependent structure, the so-called intrinsic curvature [45, 275].

Comprehensive genome analysis of DNA curvature in regulatory regions was presented in several studies [84, 23, 155, 109, 121, 120, 193]. For some bacterial genomes, it was found that regulatory regions are significantly more

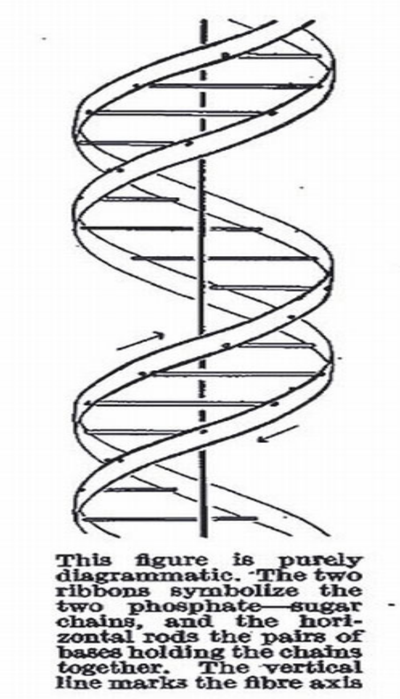

This figure is purely diagrammatic. The two ribbons symbolize the two phosphate—sugar chains, and the horizontal rods the pairs of bases holding the chains together. The vertical line marks the fibre axis

Fig. C.1 DNA Double Helix. This simple illustration was made by Odile Crick, the wife of Francis Crick.

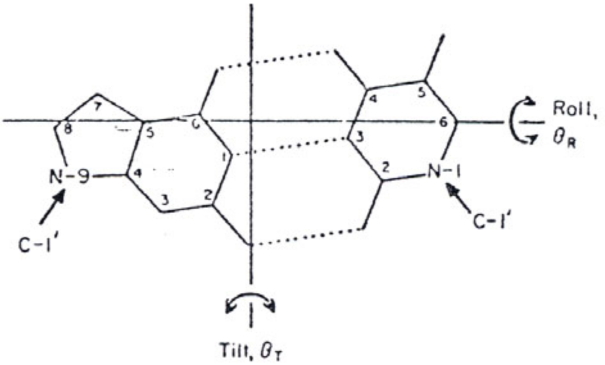


Fig. C.2 Rotation degrees of freedom of planar base pairs

curved than their neighboring coding regions and as compared to the expectations based on their dinucleotide composition. For several particular promoters of certain bacteria, it was established that DNA intrinsic curvature upstream from the promoters is related to the activity of the promoter

[200, 28, 184, 253, 88, 134]. However, the factors influencing the distribution of DNA curvature have not been clearly identified and characterized yet.

The existence of an upstream intrinsic DNA curvature has become associated with the activity of the bacterial promoter since 1984 [26]. The influence of phased A- and T-tracts (upstream curved sequences - UCS) on bacterial promoters was reported in a number of publications [200, 28, 6, 199, 205, 234]. The prediction analysis showed that the participation of curved DNA sequences at nearly each stage of the transcription process is a rule rather than an exception.

Not every function of DNA curvature is understood yet, but the DNA curvature effects have been detected in a wide variety of bacterial promoters subjected to positive or negative gene transcription control.

The conclusions that can be drawn from these kinds of small-scale characterizations of curved sequences are limited to a family of genes, a specific organism, and, at most, to close phylogenetic relatives.

The regulatory role of the sequence-dependent DNA curvature may be studied not just in specific genes, but also in a broader genomic context. The first studies of complete genome sequences [84, 121, 83] showed that E. coli promoter regions have a tendency to be more curved than E. coli coding sequences. What is the mechanism by which curved DNA sequences upstream from the core promoter activate transcription? The early model suggested that curved sequences could act as docking regions for RNA polymerase so that its local concentration in the proximity of the promoter increases [144]. However, many experiments that involved kinetics steps during the initiation of transcription suggest a far more complex scenario, in which DNA curvature plays an active role in the formation of complexes of transcription components. For some promoters, the increased isomerization rate of the open RNA polymerase-promoter complex as compared to the closed complex has been attributed to the effect of UCS. It has also been noted that both the affinity for RNA polymerase and the isomerization rate can be affected simultaneously by the presence of UCS.

Upstream curved sequences may also affect the promoter clearance. The respective function may depend on the position and the extent of the upstream contacts between the RNA polymerase and the curved sequence, e.g., additional contacts could act as a spring, helping to release the transcribing complex from the promoter. It was also demonstrated that specific proteins, the so-called transcription factors, bind to the DNA in the promoter region and change the efficiency of transcription even when they do not directly interact with RNA polymerase. Probably, many of these proteins recognize curved DNA regions rather than specific sequences. This means that even a small intrinsic region of curvature may enhance the protein-DNA binding affinity. This might lead to speculations that local curvature serves to fine-tune the interaction of promoters with regulatory factors [242, 275].

To sum up the above, curved DNA sequences commonly participate in prokaryotic gene expression and are often found within promoters and/or upstream from the promoters. In each location, curved DNA can serve different important functions in the process of transcription initiation.

C.2 History of DNA Curvature

The process of discovering DNA curvature was quite slow. Twenty years after the paper of Watson and Crick [29] on the double-helix DNA structure had been published, the evidence of the influence of base composition on the average twist between adjacent base- pairs came from DNA X-ray fiber diagrams. In 1980, Trifonov and Sussman [260] came up with the idea that non-parallel adjacent base pairs, if repeated at a distance close to the full turn of the DNA double helix, will cause the inclination of the DNA axis in the same direction and thus could facilitate DNA bending in chromatin. Indeed, it was found [260] that the spacing of certain dinucleotides (especially AA and TT) correlates with the DNA helical repeat in eukaryotes.

It was suggested that this weak periodicity might reflect a phased curving of the molecule due to the wedge-like structure of each dinucleotide (see Fig. C.3). To the best of our knowledge, this was the first publication stating the sequence-dependent nature of the intrinsic overall DNA structure. The wedge model is also called the "nearest neighbor model" since the geometry of a stack of two base pairs is defined by the two constituent nucleotides as if a "wedge" were inserted between the base pairs, the influence of more distant neighbors being ignored [266]. The term "curved DNA" was first used by Trifonov [255] with referrence to a molecule which is curvilinear rather than straight in the absence of any external forces (in contrast to "bent DNA" , which is deformed by applying certain force).

Subsequent experimental studies which employed gel-electrophoresis as well as X-ray diffraction supported the hypothesis of sequence-dependent

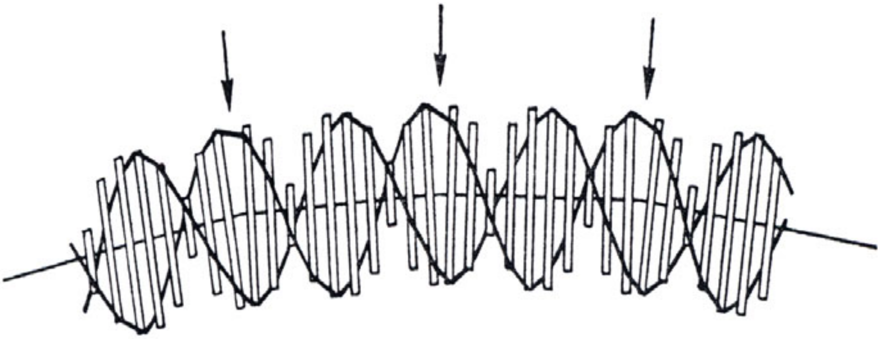


Fig. C.3 Curving of DNA molecule by periodical disposition of non-parallel (wedge-like) combinations of base-pairs (arrows). (From Trifonov, E. N. and Sussman, J.L., Proc. Natl. Acad. Sci. U.S.A., 77, 3816, 1980)

DNA structure [64, 63]. Marini et al. [181] observed anomalous behavior of DNA fragments which contained phased adenine tracts and, in line with the Trifonov's hypothesis [260, 255], attributed the anomaly to the curvature of the fragments. Hagerman's studies [97, 98] also confirmed the hypothesis that phased A-tracts produce intrinsic curvature, while eliminating alternative explanations based on unusual rigidity and/or flexibility caused by A-tracts. Hagerman and other groups clarified the principle role played by the runs of adenines in the DNA curvature [98, 288, 66, 166]. However, all these pioneer studies gave only indirect evidence of the existence of DNA curvature, which was directly confirmed in 1986 by two research groups. Griffith et al. [96] examined cloned (presumably curved) DNA fragments, 200 bp long, by means of electron microscopy and observed that they were, indeed, strongly curved (while the control fragment showed no unusual curvature). Ulanovsky et al. [265] used another method of studying curved DNA, namely, ring formation with a 21-residue synthetic duplex expected to be curved. Indeed, the diameters of the rings formed were substantially smaller than those usually observed for "straight" DNA molecules. The researches that used the method of cryo-electron microscopy, especially those of Prof. Dubochet's group [67, 70] made it possible to reconstruct the DNA trajectory in vitreous ice and led to a reasonable representation of 3-D shapes of DNA molecules [67]. By the mid-1990s, the concept of DNA curvature had become generally accepted.

C.3 Prediction of DNA Curvature

The studies presented in this section are based on the "nearest-neighbor model" or the "wedge model", which provides a realistic approximation of the intrinsic DNA trajectory [256, 234, 255, 22] and is currently widly used [134, 11, 107, 277, 92]. According to this model, the intrinsic DNA trajectory is adequately represented by a series of DNA axis deflections. It is well known from mechanics that the angular position (orientation) of an axis may be described by three Euler angles. The Euler angles were introduced by the famous mathematician Leonhard Euler to describe the orientation of a rigid body in a 3-dimensional Euclidean space. A specific orientation may be depicted by a sequence of three rotations presented by the Euler angles (see, e.g., Wolfram Mathworld http://mathworld.wolfram.com/EulerAngles.html). Different authors use different sets of angles or different names for the same angles. In [292] and later in [234, 22], a nonconventional set of Euler angles was introduced: the helical twist angle Ω, the deflection (wedge) angle σ and the direction of deflection angle δ. The nearest-neighbor set of angles consists of 26 independent values corresponding to mutual orientations of stacked base pairs. These 26 angles (10 helical twist angles, 10 deflection angles, and 6 direction-of-deflection angles) were derived from experimental data [22]. An algorithm based on the calculation of geometric transformations according to these 26 angles was used to construct a CURVATURE computer program

(Shpigelman, Trifonov, and Bolshoy [234]). The software allows to plot a sequence-dependent spatial trajectory of the DNA double helix and/or the curvature distribution along the DNA molecule. The program can be used to investigate possible roles of curvature in gene expression, for example, by locating curved portions of DNA, which may play an important role in sequence-specific protein-DNA interactions.

Let us start with some classical definitions. The osculating circle is the best circle that approximates the curve at point P. If we ignore degenerate curves such as straight lines, the osculating circle of a given curve at a given point is unique. Curvature K at point P is calculated from the formula R = $\frac{1}{|K|}$, where R is the radius of the osculating circle at point P. However, this classical curvature is not an appropriate measure of DNA intrinsic curvature [234]. Shpigelman et al. proposed to use a measure dependent on the arc size. Instead of taking the best circle that approximates the curve at point the authors used the circle best approximating a path segment of the length equal to the arc size, which is the program parameter. The DNA curvature is usually measured in DNA curvature units (cu) evaluated by Trifonov and Ulanovsky [254]:

$$\kappa = \frac{K}{K_{nucleosome}} = \frac{42.8\ \text{Å}}{\text{radius of the approximating arc}}$$

For example, a 125 bp-long segment with the shape close to a half-circle has the curvature value of about 0.34 cu. Such strongly curved fragments (cu values of more than 0.3) sometimes appear in genomic sequences.

C.4 Environmental Effects on DNA Curvature

Temperature and other environmental influences on the intrinsic DNA curvature, as manifested by electrophoretic anomalies, were studied in the 1980s - 1990s (see, e.g., [65, 268]). Ussery et al. compared the anomalous migration of 11 specifically-designed oligonucleotides in polyacrylamide gels. At low temperatures (25°C and below), most of the sequences exhibited some degree of anomalous migration. Increased temperature had a significant effect on the anomalous migration of some (curved) sequences, while a limited effect on others was observed; at 50°C, only one sequence migrated anomalously. Chan et al., using a variety of physical methods, detected a temperature-dependent "premelting" event, which eliminates DNA curvature as well. The authors suggested that this event corresponds to a specific curved DNA structure [41]. Lopez-Garcia [176] came up with the hypothesis that the distinctive DNA topology in hyperthermophilic Archaea has appeared as a result of evolution and plays an important role in gene regulation in response to environmental changes. In the context of the paper, the term "DNA topology" is almost a synonym of "DNA supercoiling". In our book, we consider the DNA curvature as a simplified feature of the DNA overall structure. Nevertheless, the conclusions of Lopez-Garcia are very close to those which we arrive at in Chapter 7. This is not surprising because DNA topology and DNA spatial trajectory are directly related.

The prokaryotes' cell components, including DNA, are influenced by the environmental conditions of their habitat. For example, it is very interesting to look at the three strains of Prochlorococcus marinus genomes from the point of view of their size. The strain that adapted to high light intensities has the smallest genome of all oxygenic phototrophs, while the strain that adapted to low light intensities has the largest genome [267].

The influence of temperature on DNA conformation might explain the distribution of DNA topoisomerases and DNA-binding proteins in extant organisms and may provide information on the early stages of evolution and temperature adaptation. Most probably, the primary reason for which the topoisomerases evolved was the release of local torsion stress generated during transcription, replication, and recombination. On the one hand, all the mesophilic Eubacteria and most of the Archaea use the gyrase enzyme to create negative DNA supercoiling in addition to local unwinding for the purpose of initiating DNA activity [184]. Such additional local unwinding can be accounted for by the UCS since curved DNA structures mimic a negative supercoil. On the other hand, hyperthermophiles use reverse gyrase to create positive supercoiling since thermophilic temperatures supply the energy of activation required for releasing DNA strands without introducing excessive denaturation. It has been universally proved that the effect of DNA curvature disappears with rising temperature. This relationship between the temperature and the DNA curvature may suggest a functional significance of the latter in mesophilic prokaryotes only.

The relationships between some other environmental conditions and DNA curvature were studied as well. Ussery et al. [268] found that Mg2+ had a strong influence on the migration of certain sequences, while spermine enhanced the anomalous migration of a different set of sequences. Sequences with the GGC motif exhibit greater curvature than it is predicted on the basis of the currently-used angles for the nearest-neighbor wedge model and are especially sensitive to Mg2+.

References

[1] Abrahamson, K.: Generalized string matching. Computing 16(6), 1039–1051 (1987)

[2] Adams, E.N.: Consensus techniques and the comparison of taxonomic trees. Systematic Zoology 21, 390–397 (1972)

[3] Adams, E.S., Meltzer, A.C.: Trigrams as index elements in full text retrieval, observations and experimental results. In: Proceedings of ACM Computer Scientific Conference. ACM Press, New York (1993)

[4] Aho, A.V.: Algorithms for finding patterns in strings. In: Handbook of Theoretical Computer Science. Algorithms and Complexity (A), vol. A, pp. 255–300 (1974)

[5] Aho, A.V.: Algorithms for finding patterns in strings. In: Handbook of Theoretical Computer Science. Algorithms and Complexity (A), vol. A, pp. 255–300 (1990)

[6] Aiyar, S.E., Gourse, R.L., Ross, W.: Upstream a-tracts increase bacterial promoter activity through interactions with the RNA polymerase alpha subunit. Proc. Natl. Acad. Sci. USA 95, 14652–14657 (1998)

[7] Akhiezer, N.: The classical moment problem and some related questions in analysis. Hafner Publishing Co., New York (1965)

[8] Alberts, B., Johnson, A.: Molecular Biology of the Cell, 4th edn. Garland Science, NY (2002)

[9] Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402 (1997)

[10] Amir, A., Lewenstein, M., Porat, E.: Faster algorithms for string matching with k mismatches. Journal of Algorithms 50, 257–275 (2004)

[11] Amit, R., Oppenheim, A.B., Stavans, J.: Increased bending rigidity of single DNA molecules by H-NS, a temperature and osmolarity sensor. Biophys. J. 84, 2467–2473 (2003)

[12] Anderson, N.L., Anderson, N.G.: Proteome and proteomics: new technologies, new concepts and new words. Electrophoresis 19(11), 1853–1861 (1998)

[13] Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: Boostmap: A method for efficient approximate similarity rankings. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), vol. 2, pp. 268–275 (2004)

[14] Athitsos, V., Sclaroff, S.: Estimating hand pose from a cluttered image. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), vol. 1, pp. 432–439 (2003)

[15] Baker, D., McCallum, A.: Distributional clustering of words for text classification. In: Croft, W.B., Moffat, A., Van Rijsbergen, C.J., Wilkinson, R., Zobel, J. (eds.) Proceedings of SIGIR 1998, 21st ACM International Conference on ReResearch and Development in Information Retrieval, Melbourne, Australia, pp. 96–103 (1998)

[16] Banfield, J.D., Raftery, A.E.: Model-based gaussian and non-gaussian clustering. Biometrics 49, 803–821 (1993)

[17] Bellman, R.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)

[18] Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. In: Pacific Symposium on Biocomputing, pp. 6–17 (2002)

[19] Ben-Hur, A., Guyon, I.: Detecting stable clusters using principal component analysis. In: Brownstein, M.J., Khodursky, A. (eds.) Methods in Molecular Biology, pp. 159–182. Humana press, Totowa (2003)

[20] Blais, B.S.: The Role of the Environment in Synaptic Plasticity: Towards an Understanding of Learning and Memory. Thesis submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy in the Department of Physics at Brown University (1998), http://web.bryant.edu/bblais/pdf/chap-introduction.pdf

[21] Bolshoy, A.: DNA sequence analysis linguistic tools: contrast vocabularies, compositional spectra and linguistic complexity. Appl. Bioinformatics 2(2), 103–112 (2003)

[22] Bolshoy, A., McNamara, P., Harrington, R.E., Trifonov, E.N.: Curved DNA without A-A: experimental estimation of all 16 DNA wedge angles. Proc. Natl. Acad. Sci. USA 88, 2312–2316 (1991)

[23] Bolshoy, A., Nevo, E.: Ecologic genomics of DNA: upstream bending in prokaryotic promoters. Genome Res. 10, 1185–1193 (2000)

[24] Bolshoy, A., Shapiro, K., Trifonov, E.N., Ioshikhes, I.: Enhancement of the nucleosomal pattern in sequences of lower complexity. Nucleic Acids Res. 25, 3248–3254 (1997)

[25] Bolshoy, A., Volkovich, Z.: Whole-genome prokaryotic clustering based on gene lengths. Discrete Applied Mathematics 157(10), 2370–2377 (2009)

[26] Bossi, L., Smith, D.M.: Suppressor sufj - a novel type of transfer-RNA mutant that induces translational frameshifting. Proc. Natl. Acad. Sci. USA 81, 6105–6109 (1984)

[27] Bourgain, J.: On Lipschitz embedding of finite metric spaces in Hilbert space. Israel Journal of Mathematics 52(1), 46–52 (1985)

[28] Bracco, L., Kotlarz, D., Kolb, A., Diekmann, S., Buc, H.: Synthetic curved DNA sequences can act as transcriptional activators in Escherichia coli. EMBO J. 8, 4289–4296 (1989)

[29] Bram, S.: Variation of type-B DNA X-ray fiber diagrams with base composition. Proc. Natl. Acad. Sci. U.S.A. 70, 2167–2170 (1973)

[30] Breckenridge, J.N.: Replicating cluster analysis: Method, consistency, and validity. Multivariate Behavioral Research 24, 147–161 (1989)

[31] Brendel, V., Beckmann, J.S., Trifonov, E.N.: Linguistics of nucleotide sequences: morphology and comparison of vocabularies. Journal of Biomolecular Structure and Dynamics 4(1), 11–21 (1986)

[32] Brinkmann, H., Philippe, H.: Archaea sister group of bacteria? Indications from tree reconstruction artifacts in ancient phylogenies. Molecular Biology and Evolution 16(6), 817–825 (1999)

[33] Brocchieri, L.: Phylogenetic inference from molecular sequences: review and critique. Theor. Popul. Biol. 59(1), 27–40 (2001)

[34] Bugaenko, N.N., Gorban, A.N., Sadovsky, M.G.: Information content of nucleotide sequences and their parts. Biofizika 42, 1047–1053 (1997)

[35] Calinski, R., Harabasz, J.: A dendrite method for cluster analysis. Commun. Statistics 3, 1–27 (1974)

[36] Cancedda, N., Gaussier, E., Goutte, C., Renders, J.: Word-sequence kernels. Journal of Machine Learning Research 3, 1059–1082 (2003)

[37] Carreira-Perpian, M.A.: A review of dimension reduction techniques. Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield (January 1997)

[38] Cavnar, W.B.: Using an n-gram-based document representation with a vector processing retrieval model. In: Proceedings of the Fourth Text Retrieval Conf., TREC-3 (1995)

[39] Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Computational Statistics and Data Analysis 14, 315 (1992)

[40] Chakravarthy, S.V., Ghosh, J.: Scale-based clustering using the radial basis function network. IEEE Transactions on Neural Networks 7(5), 1250–1261 (1996)

[41] Chan, S.S., Breslauer, K.J., Austin, R.H., Hogan, M.E.: Thermodynamics and premelting conformational changes of phased (dA)5 tracts. Biochemistry 32, 11776–11784 (1993)

[42] Chargaff, E.: Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. Experientia 6(6), 201–209 (1950)

[43] Chargaff, E., Rudner, R., Karkas, J.D.: Separation of B. subtilis DNA into complementary strands. iii. direct analysis. Proc. Natl. Acad. Sci. USA 60, 921–922 (1968)

[44] Cheng, R., Milligan, G.W.: Measuring the influence of individual data points in a cluster analysis. J. Classification 13, 315–335 (1996)

[45] Churchill, M.E.A., Jones, D.N.M., Glaser, T., Hefner, H., Searles, M.A., Travers, A.A.: HMG-D is an architecture-specific protein that preferentially binds to DNA containing the dinucleotide TG. EMBO J. 14, 1264–1275 (1995)

[46] Cohan, F.M.: What are bacterial species? An. Rev. Microbiol. 56, 457–487 (2002)

[47] Cohen, J.D.: Highlights: Language- and domain-independent automatic indexing terms for abstracting. J. of the Amer. Society for Information Sciences 46(3) (1995)

[48] Cole, R., Hariharan, R.: Approximate string matching: A faster simpler algorithm. In: Proc. 9th ACMSIAM Symposium on Discrete Algorithms (SODA), pp. 463–472 (1995)

[49] Collado-Vides, J.: A transformational-grammar approach to the study of the regulation of gene expression. J. Theor. Biol. 136(4), 403–425 (1989)

[50] Collado-Vides, J.: A syntactic representation of units of genetic informationa syntax of units of genetic information. J. Theor. Biol. 148(3), 401–429 (1991)

[51] Collado-Vides, J.: A linguistic representation of the regulation of transcription initiation.ii. distinctive features of sigma 70 promoters and their regulatory binding sites. J. Biosystems 29(2-3), 105–128 (1993)

[52] Conway, J.H., Sloane, N.J.A.: He Leech Lattice, Sphere Packings, and Related Topics. Springer, Heidelberg (1984)

[53] Cook, D., Buja, A., Cabrera, J.: Projection pursuit indices based on orthonormal function expansions. Journal of Computational and Graphical Statistics 2, 225–250 (1993)

[54] Damashek, M.: Gauging similarity with n-grams: language-independent categorization of text. Science 267, 843–848 (1995)

[55] Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM (1964)

[56] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)

[57] Delsuc, F., Brinkmann, H., Philippe, H.: Phylogenomics and the reconstruction of the tree of life. Nat. Rev. Genet. 6(5), 361–375 (2005)

[58] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likehood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B. 39, 1–38 (1977)

[59] Dhillon, I., Kogan, J., Nicholas, C.: Feature selection and document clustering. In: Berry, M. (ed.) A Comprehensive Survey of Text Mining, pp. 73–100. Springer, Heidelberg (2003)

[60] Dhillon, I.S., Subramanyam, M., Rahul, K.: Enhanced word clustering for hierarchical text classification. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2002), pp. 191–200 (2002)

[61] Dhillon, I.S., Guan, Y., Kogan, J.: Refining clusters in high-dimensional text data. In: Dhillon, I.S., Kogan, J. (eds.) Proceedings of the Workshop on Clustering High Dimensional Data and its Applications at the Second SIAM International Conference on Data Mining, pp. 71–82. SIAM, Philadelphia (2002)

[62] Diaconis, P., Freedman, D.: Asymptotics of graphical projection pursuit. Annals of Statistics 12, 793–815 (1984)

[63] Dickerson, R.E., Drew, H.R.: Kinematic model for B-DNA. Proc. Natl. Acad. Sci. USA 78, 7318–7322 (1981)

[64] Dickerson, R.E., Drew, H.R.: Structure of a B-DNA dodecamer ii. influence of base sequence on helix structure. J. Mol. Biol. 149, 761–786 (1981)

[65] Diekmann, S.: Temperature and salt dependence of the gel migration anomaly of curved DNA fragments. Nucleic Acids Res. 15, 247–265 (1987)

[66] Diekmann, S., Wang, J.C.: On the sequence determinants and flexibility of the kinetoplast DNA fragment with abnormal gel electrophoretic mobilities. J. Mol. Biol. 186, 1–11 (1985)

[67] Dubochet, J., Bednar, J., Furrer, P., Stasiak, A.Z., Stasiak, A., Bolshoy, A.: Determination of the DNA helical repeat by cryo-electron microscopy. Nat. Struct. Biol. 1, 361–363 (1994)

[68] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley and Sons, Chichester (2000)

[69] Dudoit, S., Fridlyand, J.: A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biol. 3(7) (2002)

[70] Dustin, I., Furrer, P., Stasiak, A., Dubochet, J., Langowski, J., Egelman, E.: Spatial visualization of DNA in solution. J. Struct. Biol. 107, 15–21 (1991)

[71] Espinosa-Urgel, M., Tormo, A.: Sigma(s)-dependent promoters in escherichia-coli are located in DNA regions with intrinsic curvature. Nucleic Acids Research 21, 3667–3670 (1993)

[72] Everitt, B.S., Hand, D.J.: Finite mixture distributions. Chapman and Hall, London (1981)

[73] Felsenstein, J.: Phylip (phylogeny inference package) version 3.57c. Technical report, Department of Genetics, University of Washington, Seattle (1995)

[74] Forgy, E.W.: Cluster analysis of multivariate data - efficiency vs interpretability of classifications. Biometrics 21(3), 768 (1965)

[75] Fowlkes, E.W., Mallows, C.L.: A method for comparing two hierarchical clusterings. J. Am. Stat. Assoc. 78, 553–584 (1983)

[76] Fraley, C., Raftery, A.E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. Computer Journal 41(8), 578 (1998)

[77] Frenkel, Z.M.: Does protein relatedness require sequence matching? Alignment via networks in sequence space. J. Biomol. Struct. Dyn. 26(2), 215–222 (2008)

[78] Friedman, J.H.: Exploratory projection pursuit. J. of the American Statistical Association 82(397), 249–266 (1987)

[79] Friedman, J.H., Tukey, J.W.: A projection pursuit algorithm for exploratory data analysis. IEEE Transactions on Computers 23, 881–890 (1974)

[80] Friedman, J.H., Rafsky, L.C.: Multivariate generalizations of the Wolfowitz and Smirnov two-sample tests. Annals of Statistics 7, 697–717 (1979)

[81] Frilyand, J., Dudoit, S.: Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Stat. Berkeley Tech. Report., 600 (2001)

[82] Frontali, C., Pizzi, E.: Conservation and divergence of repeated structures in plasmodium genomes: the molecular drift. Acta Leidensia 60, 69–81 (1991)

[83] Gabrielian, A., Bolshoy, A.: Sequence complexity and DNA curvature. Computers & Chemistry 38, 65–74 (1999)

[84] Gabrielian, A.E., Landsman, D., Bolshoy, A.: Curved DNA in promoter sequences. Silico Biol. 1, 183–196 (2000)

[85] Galil, Z., Giancarlo, R.: Improved string matching with k mismatches. SIGACT NEWS 62, 52–54 (1986)

[86] Galil, Z., Park, K.: An improved algorithm for approximate string matching. J. Comput. 19(6), 989–999 (1990)

[87] Galtier, N., Gascuel, O., Jean-Marie, A.: Markov models in molecular evolution. In: Nielsen, R. (ed.) Statistical Methods in Molecular Evolution, pp. 3–24. Springer, New York (2005)

[88] Gartenberg, M.R., Crothers, D.M.: Synthetic DNA bending sequences increase the rate of invitro transcription initiation at the Escherichia-coli lac-promoter. Journal of Molecular Biology 219, 217–230 (1991)

[89] Gelfand, M.S.: Genetic language: metaphore or analogy? Biosystems 30(1-3), 277–288 (1993)

[90] Gelfand, M.S., Koonin, E.V.: Avoidance of palindromic words in bacterial and archaeal genomes: a close connection with restriction enzymes. Nucleic Acids Research 25, 2430–2439 (1997)

[91] Golding, G.B., Gupta, R.S.: Protein-based phylogenies support a chimeric origin for the eukaryotic genome. Molecular Biology and Evolution 12(1), 1–6 (1995)

[92] Goni, J.R., Vaquerizas, J.M., Dopazo, J., Orozco, M.: Exploring the reasons for the large density of triplex-forming oligonucleotide target sequences in the human regulatory regions. BMC Genomics 7, 248 (2006)

[93] Gordon, A.D.: Identifying genuine clusters in a classification. Computational Statistics and Data Analysis 18, 561–581 (1994)

[94] Gordon, A.D.: Classification. Chapman and Hall /CRC, Boca Raton (1999)

[95] Graur, D., Li, W.H.: Fundamentals of Molecular Evolution, 2nd edn. Sinauer M.A. Associates, Sunderlend (2000)

[96] Griffith, J., Bleyman, M., Rauch, C.A., Kitchin, P.A., Englund, P.T.: Visualization of the bent helix in kinetoplast DNA by electron microscopy. Cell 46, 717–724 (1986)

[97] Hagerman, P.J.: Sequence dependence of the curvature of DNA - a test of the phasing hypothesis. Biochemistry 24, 7033–7037 (1985)

[98] Hagerman, P.J.: Sequence-directed curvature of DNA. Nature 321, 449–450 (1986)

[99] Hall, P.: Polynomial projection pursuit. Ann. Statist. 17, 589–605 (1989)

[100] Hartigan, J.: Statistical theory in clustering. Journal Classification 2, 63–76 (1985)

[101] Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-9910, Department of Computer Science University of California at Santa Cruz (1999)

[102] De Heer, T.: Experiments with syntactic traces in information retrieval. Information Storage Retrieval 10 (1974)

[103] Helgesen, C., Sibbald, P.R.: Palm - a pattern language for molecular biology. In: ISMB, pp. 172–180 (1993)

[104] Henze, N., Penrose, M.: On the multivariate runs test. The Annals of Statistics 27, 290–298 (1999)

[105] Hjaltason, G., Samet, H.: Properties of embedding methods for similarity searching in metric spaces. IEEE-PAMI 25(5), 530–549 (2003)

[106] Hofmann, T.: Probabilistic latent semantic indexing. In: Proc. ACM SIGIR, August 1999. ACM Press, New York (1999)

[107] Hoischen, C., Bolshoy, A., Gerdes, K., Diekmann, S.: Centromere parc of plasmid r1 is curved. Nucleic Acids Res. 32, 5907–5915 (2004)

[108] Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)

[109] Hosid, S., Bolshoy, A.: New elements of the termination of transcription in prokaryotes. J. Biomol. Struct. & Dyn. 22, 347–354 (2004)

[110] Hristescu, G., Farach-Colton, M.: Cluster-preserving embedding of proteins. Technical Report 99-50, Computer Science Department, Rutgers University (1999)

[111] Hsu, L.M., Giannini, J.K., Leung, T.W.C., Crosthwaite, J.C.: Upstream sequence activation of escherichia-coli argt promoter invivo and invitro. Biochemistry 30, 813–822 (1994)

[112] Huber, P.J.: Projection pursuit. Annals of Statistics 13, 435–475 (1985)

[113] Huffman, S.: Acquaintance: Language-independent document categorization by n-grams. In: Proceedings of the Fourth Text Retrieval Conference, TREC-4 (1996)

[114] Huffman, S., Damashek, M.: Acquaintance: A novel vector-space n-gram technique for document categorization. In: Proceedings of the Third Text Retrieval Conference, TREC-3 (1995)

[115] Hunston, S., Francis, G.: Studies in corpus linguistics, vol. 4. J. Benjamins, Amsterdam (2000)

[116] Hyverinen, A.: New approximations of differential entropy for independent component analysis and projection pursuit. In: Advances in Neural Information Processing Systems, vol. 10, pp. 273–279 (1998)

[117] Intrator, N., Cooper, L.: Objective function formulation of the bcm theory of visualcortical plasticity. Neural Networks 5, 3–17 (1992)

[118] Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)

[119] Jain, A.K., Moreau, J.V.: Bootstrap technique in cluster analysis. Pattern Recognition 20(5), 547–568 (1987)

[120] Jauregui, R., Abreu-Goodger, C., Moreno-Hagelsieb, G., Collado-Vides, J., Merino, E.: Conservation of DNA curvature signals in regulatory regions of prokaryotic genes. Nucleic Acids Res. 31, 6770–6777 (2003)

[121] Jauregui, R., O'Reilly, F., Bolivar, F., Merino, E.: Relationship between codon usage and sequence-dependent curvature of genomes. Microb. Comp. Genomics 3, 243–253 (1998)

[122] Johnson, S.C.: Hierarchical clustering schemes. Psychometrika 2, 241–254 (1967)

[123] Jones, M., Sibson, R.: What is projection pursuit? Journal of the Royal Statistical Society, ser. A 150, 1–36 (1987)

[124] Jukes, T.H., Cantor, C.R.: Evolution of Protein Molecules. Academy Press (1969)

[125] Kaji, M., Matsushita, O., Tamai, E., Miyata, S., Taniguchi, Y., Shimamoto, S., Katayama, S., Morita, S., Okabe, A.: A novel type of DNA curvature present in a clostridium perfringens ferredoxin gene: characterization and role in gene expression. Microbiology-Sgm 149, 3083–3091 (2003)

[126] Kandel, D., Matias, Y., Unger, R., Winkler, P.: Shuffling biological sequences. Discr. Appl. Math. 71, 171–185 (1996)

[127] Karlin, S.: Global dinucleotide signatures and analysis of genomic heterogeneity. Microbiol. 1(5), 598–610 (1998)

[128] Karlin, S., Campbell, A.M., Mrázek, J.: Comparative DNA analysis across diverse genomes. Annu. Rev. Genet. 32, 185–225 (1998)

[129] Karlin, S., Burge, C.: Dinucleotide relative abundance extremes: a genomic signature. Journal of Trends in Genetics 11(7), 283–290 (1995)

[130] Karlin, S., Ladunga, I.: Comparisons of eukaryotic genomic sequences. Proc. Natl. Acad. Sci. USA 91(26), 12832–12836 (1994)

[131] Karlin, S., Ladunga, I., Blaisdell, B.E.: Heterogeneity of genomes: measures and values. Proc. Natl. Acad. Sci. USA 91(26), 12837–12841 (1994)

[132] Karlin, S., Mrázek, J.: Compositional differences within and between eukaryotic genomes. Proc. Natl. Acad. Sci. USA 94(19), 10227–10232 (1997)

[133] Karlin, S., Cardon, L.R.: Computational DNA sequence analysis. Annu. Rev. Microbiol. 48, 619–654 (1994)

[134] Katayama, S., Matsushita, O., Jung, C.M., Minami, J., Kabe, A.O.: Promoter upstream bent DNA activates the transcription of the clostridium perfringens phospholipase c gene in a low temperature-dependent manner. EMBO J. 18, 3442–3450 (1999)

[135] Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. Wiley, NewYork (1990)

[136] Kay, L.E.: Who Wrote the Book of Life?: A History of the Genetic Code. Stanford University Press, Stanford (2000)

[137] Kendall, M.G.: Rank Correlation Methods. Charles Griffin & Co., Ltd., London (1970)

[138] Kimura, M.: A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. 16(2), 111–120 (1980)

[139] Kirzhner, V., Bolshoy, A., Volkovich, Z., Korol, A., Nevo, E.: Large-scale genome clustering across life based on a linguistic approach. Biosystems 81(3), 208–222 (2005)

[140] Kirzhner, V., Korol, A., Bolshoy, A., Nevo, E.: Compositional spectrum - revealing patterns for genomic sequence characterization and comparison. Physica A. 312, 447–457 (2002)

[141] Kirzhner, V., Korol, A., Nevo, E., Bolshoy, A.: A large-scale comparison of genomic sequences: one promising approach. Acta Biotheor. 51, 73–89 (2003)

[142] Kirzhner, V., Paz, A., Volkovich, Z., Nevo, E., Korol, A.: Different clustering of genomes across life using the A-T-C-G and degenerate R-Y alphabets: early and late signaling on genome evolution? Molecular Evolution 64(4), 448–456 (2007)

[143] Klebanov, L.B.: *N*-distances and their Applications. Charsel University in Prague, The Karolinum Press (2005)

[144] Klug, A., Travers, A.A.: The helical repeat of nucleosome wrapped DNA. Cell 56, 10–11 (1989)

[145] Kogan, J., Teboulle, M., Nicholas, C.: The entropic geometric means algorithm: an approach for building small clusters for large text datasets. In: Boley, D., et al. (eds.) Proceedings of the Workshop on Clustering Large Data Sets (held in conjunction with the Third IEEE International Conference on Data Mining), pp. 63–71 (2003)

[146] Konopka, A.: Plausible classification codes and local compositional complexity of nucleotide sequences. In: The Second International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis, vol. 1(6), pp. 69–87 (1993)

[147] Konopka, A.K.: Sequences and Codes: Fundamentals of Biomolecular Cryptology. In: Smith, D. (ed.) Biocomputing: Informatics and Genome Projects, pp. 119–174. Academic Press, San Diego (1994)

[148] Konopka, A.K.: Sequence complexity and composition. Nature Encyclopedia of the Human Genome 5, 217–224 (2003)

[149] Konopka, A.K.: Information theories in molecular biology and genomics. Nature Encyclopedia of the Human Genome 2, 464–469 (2005)

[150] Konopka, A.K., Owens, J.: Complexity charts can be used to map functional domains in DNA. Genetic Analysis-Biomolecular Engineering 7, 35–38 (1990)

[151] Konopka, A.K., Owens, J.: Non-contiguous patterns and compositional complexity of nucleic acid sequences. Computers and DNA 1(6), 147–155 (1990)

[152] Koonin, E.V., Makarova, K.S., Aravind, L.: Horizontal gene transfer in prokaryotes: Quantification and classification. Annu. Rev. Microbiol. 55, 709–742 (2001)

[153] Koonin, E.V., Galperin, M.Y.: Prokaryotic genomes: the emerging paradigm of genome-based microbiology. Curr. Opin. Genet. Dev. 7(6), 757–763 (1997)

[154] Kozobay-Avraham, L., Bolshoy, A., Volkovich, Z.: On clusterization of prokaryotes based on DNA curvature distribution. In: Last, M., Szczepaniak, P.S., Volkovich, Z., Kandel, A. (eds.) Advances in Web Intelligence and Data Mining: Studies in Computational Intelligence, vol. 4, pp. 361–375. Springer, Heidelberg (2006)

[155] Kozobay-Avraham, L., Hosid, S., Bolshoy, A.: Curvature distribution in prokaryotic genomes. In Silico Biol. 4, 361–375 (2004)

[156] Kozobay-Avraham, L., Hosid, S., Bolshoy, A.: Involvement of DNA curvature in intergenic regions of prokaryotes. Nucleic Acids Res. 34, 2316–2327 (2006)

[157] Kruskal, J.B.: Toward a practical method which helps uncover the structure of a set of observations by finding the line tranformation which optimizes a new index of condensation. In: Milton, R.C., Nelder, J.A. (eds.) Statistical Computation, pp. 427–440 (1969)

[158] Krzanowski, W., Lai, Y.: A criterion for determining the number of groups in a dataset using sum of squares clustering. Biometrics 44, 23–34 (1988)

[159] Kuhn, H.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97 (1955)

[160] Landau, G.M., Myers, E.W., Schmidt, J.P.: Incremental string comparison. SIAM Journal on Computing 27(2), 557–582 (1998)

[161] Landau, G.M., Vishkin, U.: Efficient string matching in the presence of errors. In: FOCS, pp. 126–136 (1985)

[162] Landau, G.M., Vishkin, U.: Fast parallel and serial approximate string matching. J. Algorithms 10(2), 157–169 (1989)

[163] Landauer, T.K., Dumais, S.T.: A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review 104, 211–240 (1997)

[164] Lange, T., Roth, V., Braun, L.M., Buhmann, J.M.: Stability-based validation of clustering solutions. Neural Computation 16(6), 1299–1323 (2004)

[165] Lee, J.H., Ahn, J.S.: Using n-grams for korean text retrieval. In: Proceedigns of the19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1996)

[166] Levene, S.D., Crothers, D.M.: Topological distributions and the torsional rigidity of DNA - a Monte-Carlo study of DNA circles. J. Mol. Biol. 189, 73–83 (1986)

[167] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)

[168] Levine, E., Domany, E.: Resampling method for unsupervised estimation of cluster validity. Neural Computation 13, 2573–2593 (2001)

[169] Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., Zhang, H.Y.: An information-based sequence distance and its application to whole mitochondrial genome phylogeny. Bioinformatics 17, 149–154 (2001)

[170] Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity. Springer, New York (1997)

[171] Li, W.: Random texts exhibit Zipf's-law-like word frequency distribution. IEEE Transactions on Information Theory 38(6), 1842–1845 (1992)

[172] Lin, J., Gerstein, M.: Whole-genome trees based on the occurrence of folds and orthologs: implications for comparing genomes on different levels. Genome Res. 10(6), 808–818 (2000)

[173] Linnik, Y., Ostrovskii, J.: Decomposition of Random Variables and Vectors. American Mathematical Society, Providence (1977)

[174] Lodhi, H., Cristianini, N., Shawe-Taylor, J., Watkins, C.: Text classication using string kernels. In: Advances in Neural Information Processing Systems, vol. 13. MIT Press, Cambridge (2001)

[175] Lodish, H., Scott, M.P.: Molecular Cell Biology, 5th edn. W.H. Freeman and Company, New York (2004)

[176] Lopez-Garcia, P.: DNA supercoiling and temperature adaptation: A clue to early diversification of life? J. Mol. Evol. 49, 439–452 (1999)

[177] Lukacs, E.: Characteristic Functions. Griffin (1970)

[178] Mantegna, R.N., Buldyrev, S.V., Goldberger, A.L., Havlin, S., Peng, C.K., Simons, M., Stanley, H.E.: Linguistic features of non-coding DNA sequences. Physical Review Letters 73, 3169–3172 (1994)

[179] Mardia, J., Kent, K., Bibby, J.: Multivariate Analysis. Academic Press, San Diego (1979)

[180] Margush, T., Mcmorris, F.R.: Consensus n-trees. Bulletin of Mathematical Biology 43, 239–244 (1981)

[181] Marini, J.C., Levene, S.D., Crothers, D.M., Englund, P.T.: Bent helical structure in kinetoplast DNA. Proc. Natl. Acad. Sci. U.S.A. 79, 7664–7668 (1982)

[182] Markov, A.A.: Primer statisticheskogo issledovaniya nad tekstom "evgeniya onegina", illyustriruyuschij svyaz ispytanij v cep. Izvestiya Akademii Nauk Ser. 6(3), 153–162 (1913)

[183] Martindale, C., Konopka, A.K.: Oligonucleotide frequencies in DNA follow a yule distribution. Computer & Chemistry 20(1), 35–38 (1996)

[184] McAllister, C.F., Achberger, E.C.: Rotational orientation of upstream curved DNA affects promoter function in bacillus-subtilis. Journal of Biological Chemistry 264, 10451–10456 (1989)

[185] McLachlan, G.J., Peel, D.: Finite Mixure Models. Wiley, Chichester (2000)

[186] McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extentions. Wiley, Chichester (1996)

[187] Mercer, J.: Functions of positive and negative type, and their connection with the theory of integral equations. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 83(559), 69–70 (1909)

[188] Milligan, G., Cooper, M.: An examination of procedures for determining the number of clusters in a data set. Psychometrika 50, 159–179 (1985)

[189] Mitchell, D., Bridge, R.: A test of chargaff's second rule. Biochem. Biophys. Res. Commun. 340(1), 90–94 (2006)

[190] Mufti, G.B., Bertrand, P., Moubarki, L.E.: Determining the number of groups from measures of cluster validity. In: Proceedings of ASMDA 2005, pp. 404–414 (2005)

[191] Nei, M., Kumar, S.: Molecular Evolution and Phylogenetics. Oxford University Press, New York (2000)

[192] Nikolaou, C., Almirantis, Y.: Deviations from chargaff's second parity rule in organellar DNA. insights into the evolution of organellar genomes. Gene 381, 34–41 (2006)

[193] Olivares-Zavaleta, N., Jáuregui, R., Merino, E.: Genome analysis of Escherichia coli promoter sequences evidences that DNA static curvature plays a more important role in gene transcription than has previously been anticipated. Genomics 87, 329–337 (2006)

[194] Packard, A.: Cephalopods and fish: The limits of convergence. Biol. Rev. 47, 241–307 (1972)

[195] Page, D.M., Holme, E.C.: Molecular Evolution: A Phylogenetic Approach. Blackwell Science Inc., Malden (1998)

[196] Van Passel, M.W., Kuramae, E.E., Luyf, A.C., Bart, A., Boekhout, T.: The reach of the genome signature in prokaryotes. J. Evol. Biol. 6, 0–84 (2006)

[197] Paz, A., Kirzhner, V., Nevo, E., Korol, A.: Coevolution of DNA-interacting proteins and genome "dialect". Mol. Biol. Evol. 23, 56–64 (2006)

[198] Pereira, F.C.N., Tishby, N., Lee, L.: Distributional clustering of english words. In: Meeting of the Association for Computational Linguistics, pp. 183–190 (1993)

[199] Perez-Martin, J., de Lorenzo, V.: Clues and consequences of DNA bending in transcription. Annu. Rev. Microbiol. 51, 593–628 (1997)

[200] Perez-Martin, J., Rojo, F., de Lorenzo, V.: Promoters responsive to DNA bending: a common theme in prokaryotic gene expression. Microbiol. Rev. 58, 268–290 (1994)

[201] Philippe, H., Delsuc, F., Brinkmann, H., Lartillot, N.: Phylogenomics. Annu. Rev. Ecol. Evol. Syst. 36, 541–562 (2005)

[202] Pietrokovski, S., Trifonov, E.N.: Imported sequences in the mitochondrial yeast genome identified by nucleotide linguistics. J. of Molecular Evolution 50, 129–137 (1992)

[203] Pietrokovski, S., Hirshonn, J., Trifonov, E.N.: Linguistic measure of taxonomic and functional relatedness of nucleotide sequences. Journal of Biomolecular Structure and Dynamics 7(6), 1251–1268 (1990)

[204] Pizzi, E., Frontali, C.: Divergence of noncoding sequences and of insertions encoding nonglobular domains at a genomic region well conserved in plasmodia. J. Mol. Evol. 50, 474–480 (2000)

[205] Plaskon, R.R., Wartell, R.M.: Sequence distributions associated with DNA curvature are found upstream of strong E. coli promoters. Nucleic Acids Res. 15, 785–796 (1987)

[206] Popov, O., Segal, D.M., Trifonov, E.N.: Linguistic complexity of protein sequences as compared to texts of human languages. Biosystems 38, 65–74 (1996)

[207] Pride, D.T., Meinersmann, R.J., Wassenaar, T.M., Blaser, M.J.: Evolutionary implications of microbial genome tetranucleotide frequency biases. Genome Research 13(2), 145–158 (2003)

[208] Pride, D.T., Wassenaar, T.M., Ghose, C., Blaser, M.J.: Evidence of host-virus co-evolution in tetranucleotide usage patterns of bacteriophages and eukaryotic viruses. BMC Genomics 18, 7–8 (2006)

[209] Rachev, S.T.: Probability Metrics and the Stability of Stochastic Models. Wiley, New York (1991)

[210] Rand, W.: Objective criteria for the evaluation of clustering methods. Journal Am. Stat. Assoc. 66, 846–850 (1971)

[211] Reinert, G., Schbath, S., Waterman, M.S.: Probabilistic and statistical properties of words: an overview. Journal of Comput. Biology 7(1), 1–46 (2000)

[212] Robertson, A.M., Willett, P.: Searching for historical word-forms in a database of 17th century english text using spelling-correction methods. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1992)

[213] Robins, H., Krasnitz, M., Barak, H., Levine, A.: A relative-entropy algorithm for genomic fingerprinting captures host-phage similarities. J. Bacteriol. 187, 8370–8374 (2005)

[214] Rogozin, I.B., Makarova, K.S., Murvai, J., Czabarka, E., Wolf, Y.I., Tatusov, R.L., Szekely, L.A., Koonin, E.V.: Connected gene neighborhoods in prokaryotic genomes. Nucleic Acids Res. 30(10), 2212–2223 (2002)

[215] Rose, K., Gurewitz, E., Fox, G.: Statistical mechanics and phase transitions in clustering. Physical Review Letters 65(8), 945–948 (1990)

[216] Roth, V., Lange, V., Braun, M., Buhmann, J.: A resampling approach to cluster validation. In: COMPSTAT (2002), `http://www.cs.uni-bonn.De/~braunm`

[217] Rousu, J., Shawe-Taylor, J.: Efficient computation of gap-weighted string kernels on large alphabets

[218] Russell, G.J., Walker, P.M., Elton, R.A., Subak-Sharpe, J.H.: Doublet frequency analysis of fractionated vertebrate nuclear DNA. J. Mol. Biol. 108(1), 1–23 (1976)

[219] Sadovsky, M.: Ob informacionnoi emkosti simvolnih posledovatelnostei. J. Vichislitelnie technologii 10, 82–90 (2005)

[220] Sadovsky, M.G.: Information capacity of nucleotide sequences and its applications. Bulletin of Mathematical Biology 68(4), 785–806 (2006)

[221] Salamon, P., Konopka, A.: A maximum-entropy principle for the distribution of local complexity in naturally-occurring nucleotide-sequences. Computers & Chemistry 16, 117–124 (1992)

[222] Salamon, P., Wootton, J.C., Konopka, A.K., Hansen, L.K.: On the robustness of maximum-entropy relationships for complexity distributions of nucleotide-sequences. Computers & Chemistry 17, 135–148 (1993)

[223] Salton, G.: The SMART Retrieval System. Prentice-Hall, Englewood Cliffs (1971)

[224] Schbath, S.: An efficient statistic to detect over- and under-represented words in DNA sequences. J. Computational Biology 4(2), 189–192 (1997)

[225] Schbath, S., Prum, B., De Turckheim, E.: Exceptional motifs in different markov chain models for a statistical analysis of DNA sequences. J. Computational Biology 2(3), 417–437 (1995)

[226] Schmitt, A.O., Ebeling, W., Herzel, H.: The modular structure of informational sequences. Biosystems 37, 199–210 (1996)

[227] Scholkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)

[228] Schrödinger, E.: What is life? Cambridge University Press, Cambridge (1944)

[229] Scott, D.W., Thompson, J.R.: Probability density estimation in higher dimensions. In: Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface, pp. 173–179. North–Holland, Amsterdam (1983)

[230] Searls, D.B.: Linguistic approaches to biological sequences. J. Comput. Appl. Biosci. 13(4), 333–344 (1997)

[231] Searls, D.B.: The language of genes. Nature 420, 211–217 (2002)

[232] Searls, D.B.: Linguistics: Trees of life and of language. Nature 426, 391–392 (2003)

[233] Shannon, C.E.: A mathematical theory of communication. Bell System Technical J. 27, 379–423 (1948)

[234] Shpigelman, E.S., Trifonov, E.N., Bolshoy, A.: Curvature - software for the analysis of curved DNA. Comput. Appl. Biosci. 9(4), 435 (1993)

[235] Sibbald, P.R., Banerjee, S., Maze, J.: Calculating higher order DNA sequence information measures. J. Theor. Biol. 136, 475–483 (1989)

[236] Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: NIPS-12 (1999)

[237] Slonim, N., Tishby, N.: Document clustering using word clusters via the information bottleneck method. In: Research and Development in Information Retrieval, pp. 208–215 (2000)

[238] Snel, B., Huynen, M.A., Dutilh, B.E.: Genome trees and the nature of genome evolution. Annu. Rev. Microbiol. 59, 191–209 (2005)

[239] Snel, B., Bork, P., Huynen, M.: Genome phylogeny based on gene content. Nature Genet. 21(1), 108–110 (1999)

[240] Still, S., Bialek, W.: How many clusters? An information-theoretic perspective. Neural computation 16(12), 2483–2506 (2004)

[241] Sugar, C.A., James, G.M.: Finding the number of clusters in a dataset: An information-theoretic approach. J. of the American Statistical Association 98(463), 750 (2003)

[242] Suzuki, M., Yagi, N.: Stereochemical basis of DNA bending by transcription factors. Nucl. Acids Res. 23, 2083–2091 (1995)

[243] Tatusov, R.L., Natale, D.A., Garkavtsev, I.V., Tatusova, T.A., Shankavaram, U.T., Rao, B.S., Kiryutin, B., Galperin, M.Y., Fedorova, N.D., Koonin, E.V.: The cog database: new developments in phylogenetic classification of proteins from complete genomes. Nucleic Acids Res. 29(1), 22–28 (2001)

[244] Teeling, H., Meyerdierks, A., Bauer, M., Amann, R., Glockner, F.O.: Application of tetranucleotide frequencies for the assignment of genomic fragments. J. Environ. Microbiol. 6(9), 938–947 (2004)

[245] Teeling, H., Waldmann, J., Lombardot, T., Bauer, M., Oliver, F., Glockner, F.O.: Tetra: A web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. J. Bioinformatics 5, 163 (2004)

[246] Tekaia, F., Lazcano, A., Dujon, B.: The genomic tree as revealed from whole proteome comparisons. Genome Res. 9, 550–557 (1999)

[247] Tekaia, F., Yeramian, E.: Genome trees from conservation profiles. PLoS Comput. Biol. 1(7), 604–616 (2005)

[248] Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters via the gap statistic. J. Royal Statist. Soc. B 63(2), 411–423 (2001)

[249] Tibshirani, R., Walther, G.: Cluster validation by prediction strength. Journal of Computational & Graphical Statistics 14(3), 511–528 (2005)

[250] Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, pp. 368–377 (1999)

[251] Tishby, N., Slonim, N.: Data clustering by markovian relaxation and the information bottleneck method. In: NIPS, pp. 640–646 (2000)

[252] Topchy, A.P., Minaei-Bidgoli, B., Punch, W.F.: Ensembles of partitions via data resampling. In: ITCC (2), pp. 188–192 (2004)

[253] Travers, A.A.: Why bend DNA. Cell 60, 177–180 (1990)

[254] Trifonov, E.N., Ulanovsky, L.E.: Inherently curved DNA and its structural elements. In: Wells, R.D., Harvey, S.C. (eds.) Unusual DNA Structures, pp. 173–187. Springer, Berlin (1987)

[255] Trifonov, E.N.: Sequence-dependent deformational anisotropy of chromatin DNA. Nucleic Acids Res. 8, 4041–4053 (1980)

[256] Trifonov, E.N.: Curved DNA. CRC Critical Reviews in Biochemistry 19, 89–106 (1985)

[257] Trifonov, E.N.: The multiple codes of nucleotide sequences. J. Bull. Math. Biol. 51(4), 417–432 (1989)

[258] Trifonov, E.N.: Making sense of the human genome. Structure and Methods: Human Genome Initiative and DNA Recombination 1(6), 69–77 (1990)

[259] Trifonov, E.N.: DNA in profile. Trends in Biochemical Sciences 16, 467–470 (1991)

[260] Trifonov, E.N., Sussman, J.L.: The pitch of chromatin DNA is reflected in its nucleotide sequence. Proc. Natl. Acad. Sci. USA 77, 3816–3820 (1980)

[261] Troyanskaya, O.G., Arbell, O., Koren, Y., Landau, G.M., Bolshoy, A.: Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity. Bioinformatics 18(5), 679–688 (2002)

[262] Ukkonen, E.: Algorithms for approximate string matching. J. Information and Control 64, 100–118 (1985)

[263] Ukkonen, E.: On-line construction of suffix trees. Algorithmica 14, 249–260 (1995)

[264] Ulam, S.: Some combinatorial problems studied experimentally on computing machines. Academic Press, London (1972)

[265] Ulanovsky, L., Bodner, M., Trifonov, E.N., Choder, M.: Curved DNA - design, synthesis, and circularization. Proc. Natl. Acad. Sci. USA 83, 862–866 (1986)

[266] Ulanovsky, L.E., Trifonov, E.N.: Estimation of wedge components in curved DNA. Nature 326, 720–722 (1987)

[267] Ussery, D.W., Hallin, P.F.: Genome update: length distributions of sequenced prokaryotic genomes. Microbiology-Sgm 150, 513–516 (2004)

[268] Ussery, D.W., Higgins, C.F., Bolshoy, A.: Environmental influences on DNA curvature. J. Biomol. Struct. Dyn. 16, 811–823 (1999)

[269] van Belkum, A.: Short sequence repeats in microbial pathogenesis and evolution. Cell. Mol. Life Sci. 56, 729–734 (1999)

[270] van Belkum, A., Scherer, S., van Alphen, L., Verbrugh, H.: Short-sequence DNA repeats in prokaryotic genomes. Microbiol. Mol. Biol. Rev. 62, 275–293 (1998)

[271] van Belkum, A., van Leeuwen, W., Scherer, S., Verbrugh, H.: Occurrence and structure-function relationship of pentameric short sequence repeats in microbial genomes. Res. Microbiol. 150, 617–626 (1999)

[272] Vinga, S., Almeida, J.: Alignment-free sequence comparison – a review. Bioinformatics 19(4), 513–523 (2003)

[273] Volkovich, V.: On the Levy-Khintchine formulas from the generalized function point of view. In: Transactions of XXIV International Seminar on Stability Problems for Stochastic Models, Jurmala, Latvia, September 10-17, pp. 74–76 (2004)

[274] Volkovich, Z., Kirzhner, V., Bolshoy, A., Korol, A., Nevo, E.: The method of n-grams in large-scale clustering of DNA texts. Pattern Recognition 38(11), 1902–1912 (2005)

[275] Wagner, R.: Transcription Regulation in Prokaryotes. Oxford University Press Inc., New York (2000)

[276] Wang, H., Benham, C.J.: Promoter prediction and annotation of microbial genomes based on DNA sequence and structural responses to superhelical stress. BMC Bioinformatics 7, 248 (2006)

[277] Wang, L.S., Warnow, T., Moret, B.M.E., Jansen, R.K., Raubeson, L.A.: Distance-based genome rearrangement phylogeny. J. Mol. Evol. 63, 473–483 (2006)

[278] Waterman, M.S.: Introduction to Computational Biology. Chapman & Hall, London (1995)

[279] Waterman, M.S., Smith, T.F., Beyer, W.A.: Some biological sequence metrics. Advances in Mathematics 20(3), 367–387 (1976)

[280] Watkins, C.: Dynamic alighment kernels. Technical report, Technical report, UL Royal Holloway (1999)

[281] Watson, J.D., Crick, F.H.C.: Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. J-Nature 171(4356), 737–738 (1953)

[282] Watson, J.D., Baker, T.A.: Molecular Biology of the Gene, 5th edn. Benjamin Cummings, New York (2003)

[283] Winkler, W.E.: The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau (1999)

[284] Woese, C.R.: Bacterial evolution. Microbiol. Rev. 51(2), 221–271 (1987)

[285] Woese, C.R., Kandler, O., Wheelis, M.L.: Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. Proc. Natl. Acad. Sci. USA 87, 4576–4579 (1990)

[286] Wolf, Y.I., Rogozin, I.B., Grishin, N.V., Koonin, E.V.: Genome trees and the tree of life. Trends Genet. 18(9), 472–479 (2002)

[287] Wolf, Y.I., Rogozin, I.B., Grishin, N.V., Tatusov, R.L., Koonin, E.V.: Genome trees constructed using five different approaches suggest new major bacterial clades. BMC Evol. Biol. 1, 8 (2001)

[288] Wu, H.M., Crothers, D.M.: The locus of sequence-directed and protein induced DNA bending. Nature 308, 509–513 (1984)

[289] Yockey, H.P.: Information Theory, Evolution, and the Origin of Life. Cambridge University Press, Cambridge (2005)

[290] Zech, G., Aslan, B.: New test for the multivariate two-sample problem based on the concept of minimum energy. The Journal of Statistical Computation and Simulation 75(2), 109–119 (2005)

[291] Zhang, C.T., Zhang, R., Ou, H.Y.: The z curve database: a graphic representation of genome sequences. Bioinformatics 19(5), 593–599 (2003)

[292] Zhurkin, V.B., Lysov, Y.P., Ivanov, V.I.: Anisotropic flexibility of DNA and the nucleosomal structure. Nucleic Acids Res. 6, 1081–1096 (1979)

[293] Zipf, G.K.: The Psychobiology of Language. Houghton-Mifflin, Boston (1935)

[294] Zolotarev, V.M.: Modern Theory of Summation of Random Variable. Brill Academic Publishers, Leiden (1997)

[295] Zuckerkandl, E., Pauling, L.: Molecules as documents of evolutionary history. Biosystems 8(2), 357–366 (1965)

Index