字符序列的解析数论模型及其在生物信息学中的应用

Analytic Number Theory Model for character sequence and its application in bioinformatics

(申请硕士学位)

专　　业	生物物理学
研 究 生	马彬广
指导教师	张春霆 教授

天津大学理学院

2003 年 12 月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 天津大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：马彬广　　签字日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：马彬广　　导师签名：张春霆

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

摘要

生物信息学领域内的许多课题可以抽象成字符序列处理问题，例如，基因识别、蛋白质二级结构预测等。字符序列所能提供的信息不外乎来自两个方面：组成和排列。组成方面的信息可以用常规的频率去反映。问题的关键是如何反映字符序列的排列信息。本文在综述了现有算法的基础上，尝试着从数论的角度来看待字符序列分析问题，提出了字符序列的解析数论模型。在该模型中，把字符序列看成是数的表示，从而把字符序列分析问题转化成一个数论问题，并用数学分析方法辅助解决。

字符序列解析数论模型的核心概念是对偶描述子，因此，该模型有时也称作“对偶描述子方法”。对偶描述子由两部分组成：组成权重因子和位置权重函数。“组成权重因子”来源于自然计数制中“基数”的概念，是它在实数域上的推广。位置权重函数的概念则是自然数系统中所固有的，它也被推广到了实数域。为了逼近位置权重函数，傅里叶变换、小波变换等理论很自然地被引入到字符序列的处理中来。

本文给出了一种基于一定的数据集，来训练对偶描述子的交替式学习方法。训练所得的对偶描述子，就携带了原数据集的特征信息。通过本文提供的 D 值阈值判别方法，它可以用于字符序列的识别。同时，由于位置权重函数的引入，实现了位置加权统计，由此所得的结果，便是“带位置权重的频率”，简称“加权频率”。加权频率优于常规频率的地方就在于，它不仅可以反映字符序列的组成信息，还能反映它的排列信息。因此，加权频率可以作为字符序列的特征量。有了它，对偶描述子就可以和其他的一些判别方法，比如 Fisher 判别等，联合使用来进行字符序列的识别了。

以 DNA 序列分析为例，本文演示了对偶描述子在生物信息学中的应用。具体内容包括：序列特征的提取，对偶描述子学习过程的演示，D 值阈值判别和加权频率 Fisher 判别在原核生物基因识别和真核生物外显子识别中的应用。

关键词：字符序列的解析数论模型;　对偶描述子; 加权频率;　D 值阈值判别

ABSTRACT

Many tasks in bioinformatics can be abstracted as the problem of character sequence analysis, such as gene recognition, protein secondary structure prediction etc. All the information a character sequence can provide is no more than two aspects: composition and permutation. The composition information can be represented by ordinary frequencies. The key of the problem is how to reflect the permutation information of a character sequence. Based on a review of already known algorithms, a new model called **Analytic Number Theory Model** for character sequence is proposed in this dissertation from the visual angle of number theory. In this model, a character sequence is treated as a representation of a number, so that the analysis of character sequence is transformed into a problem of number theory, and to be solved by the aid of mathematical analysis.

The core concept of ANTM is **Dual Descriptor**, so it is also called **Dual Descriptor Method** sometimes. Dual descriptor is composed of two parts: **Composition Weight Factor** and **Position Weight Function**. Composition weight factor is derived from the concept of "radix" in natural number systems, and to be the generalization of it in real number field. Position weight function is an intrinsic concept of natural number systems, and it is also generalized into real number field. To approximate position weight function, Fourier transforms, wavelet transform and such theories are naturally introduced into the field of character sequence analysis.

An iterative method is posed in the dissertation for the training of dual descriptor based on a data set. The trained dual descriptor carries the information of original data set. It can be used to recognize character sequences by a method posed in this dissertation called **D-value threshold discriminant approach**. At the same time, because of the introducing of position weight function, **counting with position weight** is implemented, and the result of it is **Frequencies with Position Weight**, also call **Weighted Frequencies** for short. The advantages of weighted frequencies which make them outgo ordinary frequencies are their ability to carry both composition and permutation information. Therefore, weighted frequencies can serve as characteristic

variables of character sequences. With them, dual descriptor can be used in the combination with other approaches such as Fisher discriminant algorithm for the recognition of character sequences.

The application of dual descriptor in bioinformatics is demonstrated in this dissertation with the example of DNA sequence analysis. The content of it include: sequence feature extraction, the demonstration of the study process of dual descriptor, the application of D-value threshold discriminant approach and **weighted frequencies Fisher discriminant algorithm** in the recognition of protein coding regions in both prokaryotic and eukaryotic species.

Keywords: **Analytic Number Theory Model** for character sequence; **Dual Descriptor, Weighted Frequencies;D-value threshold discriminant approach**

目录

第一章　问题的引出

科学的发展和技术的进步是相互促进的。科学的发展为技术创新奠定知识和理论基础，而技术的进步则为科学研究提供新的工具和手段。近年来，得益于基因组测序技术的进步，人类基因组计划和其他高等生物以及微生物的全基因组测序成为现实。结合此前的蛋白质测序技术和物质结构解析技术，GenBank/EMBL/DDBJ 等国际公共数据库中的核酸序列数据、SWISS-PROT 等数据库中的氨基酸序列数据和 PDB 中的蛋白质三维结构数据与日俱增。然而，数据并不等于信息，更不等于知识，但却是信息和知识的源泉。数据经过加工处理，去芜存菁之后，才变成对人们有用的信息；信息经过提炼升华，转化成简单明白的原则以指导人们实践时，才变成理论知识。研究如何实现这一过程的学科，叫做数据挖掘或知识发现。与正在以指数方式增长的生物学数据相比，人类相关知识的增长却十分缓慢。一方面是海量的数据，另一方面是我们在医学、药物、农业和环保等方面对新知识的渴求，以期改善自身生存环境和提高生活质量，这就构成了一个巨大的矛盾。这一矛盾极大地推动了生物医学领域中知识发现的研究。我们正目睹着生物医学研究领域中的一个巨大的变革，即从传统的对单个基因、单个蛋白质的研究过渡到系统地对整个生物体的基因组学、蛋白质组学、转录组学的研究；而研究方法也从传统的观察和实验为主，过渡到数学，信息科学，计算机科学等学科的理论和方法相结合。这个变革使得大量的数理科学工作者自然地转入到生命科学的研究领域中来，使得生物信息学这个新学科应运而生。从 80 年代末诞生以来，生物信息学以基因组信息学为核心，在分析基因数据、寻找新基因、分析和预测蛋白质结构功能、分子进化、药物设计等方面发挥了实验所不能取代的巨大作用[1]。

1.1 生物测序工程与生物序列数据库

1.1.1 基因组测序与 DNA 序列数据库

早期的 DNA 测序是针对单个基因的，主要关注那些引起人类疾病的 DNA 片断。20 世纪五十到七十年代间，积累了一些零星的 DNA 序列数据。后来发现，某些疾病的产生不是单一基因突变的结果，而往往是多个基因的共同作用所致。于是，就有了完全测定整个人类基因组的提议，这就是后来出现的被称为“生命阿波罗计划”的人类基因组计划。1990 年，人类基因组组织(HUGO)和美国国家健康研究所(NIH)向美国国会提交了美国人类基因组计

划联合项目的 5 年计划，标志着人类基因组计划 15 年进程的开始。这一耗资 30 亿美元的工程旨在阐明人类基因组 30 亿个碱基对的序列，发现所有的基因及其在染色体上的位置，弄清每种基因制造的蛋白质及其作用，破译全部遗传信息，使人类第一次在分子水平上全面地认识自我。1999 年 9 月中国获准加入人类基因组计划，并于 2000 年 4 月，完成了 1%的人类基因组的工作框架图。2001 年 2 月 12 日，人类基因组图谱及初步分析结果首次公布，覆盖率达到 95%，平均测序精度 99.96 %[2]。至 2003 年，DNA 双螺旋发现 50 周年，国际人类基因组测序协作组宣布人类基因组序列图绘制成功。

伴随着人类基因组计划的实施，平行地进行一些微生物、植物、动物等模式生物基因组的研究，可为人类基因组研究做方法学和组织工作方面的准备，这些研究被称为“模式生物基因组计划”。模式生物基因组计划最初确定的模式生物有：大肠杆菌、酵母、拟南芥、线虫、果蝇和小鼠等共六种，后来逐渐加入了一些其它种类的模式生物，如河豚鱼、斑马鱼等。此外，一些具有重要生产价值的农作物和畜禽虫类，如水稻、小麦、家猪、家鸡、家蚕等的基因组也加入到测序计划中来。微生物的基因组相对较小，又与医疗健康事业密切相关，因此，微生物基因组的测序大量进行。要测序的微生物基因组主要包括细菌、古细菌等原核生物，还包括真菌等低等真核生物。这部分测序工作，称为微生物基因组计划(MGP)，是美国能源部在 1986 年启动的。人类历史上第一个完全测序的生物，就是一种微生物——嗜血流感杆菌[3]。

根据数据库 GOLD (Genomes OnLine Database)的统计[4]，截止到 2003 年 9 月 25 日，已完成了约 160 多种自由生物体全基因组 (包括 4 条染色体) 测序工作，其中包括 17 个古细菌，127 个细菌和 20 个真核生物。此外，至少 410 多个重要的人类病原体、具生物学意义与潜在经济价值的微生物基因组将被测序。目前，测序完成的大量 DNA 序列数据，主要存放在 GenBank/EMBL/DDBJ 等国际公共核酸序列数据库中，并公布在互联网上，让全世界的相关研究人员共享。

1.1.2 蛋白质测序与氨基酸序列数据库

蛋白质的测序要早于基因组测序。自从 1951 年，Sanger 和 Tuppy 等人发明了蛋白质测序方法以后，一些对分子进化感兴趣的研究者开始收集数据并进行比较研究。其中，在华盛顿特区的美国国家生物医学研究基金会工作的 Margaret Dayhoff 是最突出的一个。从 1968 年到 1978 年间，她将收集到的序列发表于系列《蛋白质序列与结构图集》(Atlas of Protein

Sequences and Structures）。基于序列相似性的蛋白质“家族”和“超家族”的概念也是从她（及其合作者）的工作中产生的。后来，又从观察到的氨基酸的突变（替代）频率汇集而成“MDM78”、“PAM250”等突变矩阵[5]。1980 年前后，在竞争建立美国国家 DNA 数据库期间，这些收集的数据被计算机化并被称为“NBRF 蛋白质序列数据库”。“蛋白质信息资源数据库”（Protein Information Resource, PIR）就源于 NBRF 数据库，并于 1984 年在 NIH 的支持下建立。从 1988 年起，PIR 库与德国慕尼黑蛋白质序列信息中心（Munich Information Center for Protein Sequences, MIPS）和日本国际蛋白质序列数据库（Japanese International Protein Sequence Database, JIPSD）合作，共同建立了“PIR 国际蛋白质序列数据库”（PIR-International Protein Sequence Database）[6]。

目前，许多蛋白质序列数据库都依赖于核酸序列数据库，因为大多数蛋白质序列是通过 DNA 测序后翻译得到的。“SWISS-PROT”是当前在数据质量方面最好的蛋白质序列数据库。它是一个增值型数据库，于 1986 年由瑞士的日内瓦大学创建。从 1987 年起，SWISS-PROT 与欧洲分子生物学实验室(European Molecular Biology Laboratory, EMBL)合作，把“EMBL 翻译核酸数据库”（TREMBL）作为它的补充。经过多年的积累，它包含了丰富的规整好的注释信息和大量与其他数据库的链接。 “SWISS-PROT” 数据库以其冗余度低，对数据的人工审读严格等特点，成为目前蛋白质序列搜寻和比较的起点[7]。

1.2 分子生物学的一些基础知识

下面叙述一些本论文要用到的分子生物学方面的基础知识。

细胞 细胞是生命的基本单位。在形态上细胞大体上由细胞核、细胞质及细胞膜三部分组成。根据细胞核的区别，生物可以分为**原核生物**和**真核生物**两大类。其中原核生物的核物质没有被一层核膜所包围，而是散布在细胞质中，没有形成明显的细胞核，因此才被称之为原核生物。原核生物主要包括细菌和古细菌。

DNA（脱氧核糖核酸）是细胞核内的遗传物质，DNA 的基本单位是核苷酸，不同的核苷酸是通过所含的**碱基**来区分的。DNA 包含的碱基有四种，即腺嘌呤、鸟嘌呤、胞嘧啶、胸腺嘧啶，分别用字母 A、G 、C、T 表示。多个核苷酸排列聚合形成多聚核苷酸，再由多聚核苷酸形成 DNA 大分子。1953 年，Watson 和 Crick 提出了 DNA 的双螺旋结构模型，认为 DNA 分子是由两条碱基互补的单链反向平行围绕中心轴而形成的双螺旋结构。**碱基互补**是指 A↔T，G↔C 配对形成氢键，其中 A 与 T 形成两个氢键，G 与 C 形成三个氢键。这种碱

基互补配对被称为 Watson-Crick 配对。两条链也分别被称为 W 链(正链)和 C 链（负链)。

基因 DNA 分子往往可以划分成不同的区域。不同区域有不同的生物学功能，其中有的区域编码特定的蛋白质或 RNA，就是通常所说的（狭义的）**基因**。基因作为遗传的基本单位，是指编码一个蛋白质或一个 RNA 分子的完整的 DNA 片断。在原核生物中，基因由一段连续的 DNA 序列构成。真核生物中，基因内部还可能有插入序列，这些插入的 DNA 序列并不用来编码蛋白质，称之为**内含子**。而其它用来编码的部分则称之为**外显子**。**ORF**(open reading frame)是和基因密切相关的一个概念，它是指以 ATG(或 GTG、TTG、CTG)开始并以和 ATG（或它们）同相位的 TGA(或 TAA、TAG)结束的一段连续的 DNA 序列。形成 ORF 结构需满足如下两个条件：（1）序列长度是 3 的整数倍；（2）在序列内部，和起始密码子 ATG 同相位的位置上不能出现终止密码子 TAG。明显地，原核生物的基因都满足 ORF 结构；而对于真核生物的基因，除非包含的所有内含子的长度和正好是 3 的整数倍并且内含子内部和起始密码子同相位的位置上不出现终止密码子，否则不满足 ORF 结构。

由于基因是以 3 个碱基为 1 组编码一个氨基酸，基因序列就具有 3 碱基的周期性。因此我们把一条基因序列中的各位置分成 3 个**相位**：和第 1、4、7、…个碱基对应的位置称之为第 1 相位；和第 2、5、8、…个碱基对应的位置称之为第 2 相位；和第 3、6、9、…个碱基对应的位置称之为第 3 相位。第 1、2、3 相位也叫做 3 个**密码子位**。

基因组 遗传学科对于**基因组**的定义发生了如下变化：经典遗传学把基因组定义为所有基因的总和；而细胞遗传学则定义为一个细胞内所有染色体的总和；分子遗传学则把基因组定义为所有 DNA 分子的总和，也就是说包括细胞核基因组和核外遗传物质的基因组。在本论文中涉及到的都是细胞核内染色体上的 DNA 分子。 基因组的 **GC 含量**则是指染色体上 DNA 序列中鸟嘌呤（G）和胞嘧啶（C）这两种碱基在四种碱基中所占的百分比。

分子生物学的中心法则 基因所携带的遗传信息体现在 DNA 序列中，四种核苷酸的不同排列组合方式上。基因指导蛋白质的合成，是通过先将 DNA 分子转录成与其互补的 RNA 分子，即 **mRNA**，然后将 mRNA 序列，每三个相连的核苷酸决定一个氨基酸，翻译成氨基酸序列（多肽），再经过折叠、转运等过程，形成在特定位置发挥特定功能的蛋白质分子。**遗传密码**就是指 mRNA 上每三个相连的核苷酸组成的三联体密码。密码子共有 64 种，其中 61 种构成氨基酸，其它三种(TAA、TAG、TGA)则为终止密码子，决定基因编码的结束。基因上的核苷酸序列与蛋白质上的氨基酸序列的关系就是通过遗传密码子来体现的。以上这种遗传信息从 DNA 到 mRNA 再到蛋白质的流动过程称之为**分子生物学的中心法则**。

蛋白质的结构层次 蛋白质的结构通常分为六个级别：一级结构，二级结构，超二级结

构，三级结构，四级结构及分子缔合体。一级结构指的是蛋白质序列；二级结构为蛋白质中多肽主链的规则排布；超二级结构为二级结构单元间的组合方式；三级结构指蛋白质的三维空间结构；四级结构则是指蛋白质亚基间的相互作用。这种分级方式，有时假定了多肽链中非邻近的氨基酸残基间相互不影响（实际上并非如此）[8]。

蛋白质二级结构定义 蛋白质的二级结构是从X射线晶体衍射或核磁共振等物理手段解析出来的蛋白质分子三维结构中的原子坐标出发，通过考察主链C^{α}碳原子的邻接位置关系而定义的蛋白质的主链结构。基于氢键模式和几何特征定义的“蛋白质二级结构辞典”（DSSP）是应用最广泛的一种蛋白质二级结构定义[9]。它定义了 8 种类型的二级结构，分别为 H (α-helix 或 4-helix)、B（β-bridge）、E（β-strand）、G（3_{10}-helix）、I（π-helix 或 5-helix）、T（β-turn）、S（bend）、C 或_（Coil）。其中，I 极少出现。另外，还有两种比较常用的定义方式：DEFINE[10]和 STRIDE[11]。上述几种定义方式，无论是在定义的类型上还是在各类型的含量上，都有所差异[12]。在蛋白质二级结构预测中，通常将上述 8 种类型归并为 3 种 E、H、C。常用的归并方法有两种：(1) E 和 B 归为 E，G 和 H 归为 H，其余归为 C；(2) E 归为 E，H 归为 H，其余归为 C。而这两种方法中，又以第一种方法最为常用。

1.3 粗粒化与生物序列分析

1.3.1 粗粒化与字符表示

关于粗粒化与符号描述，郝柏林先生有一段精彩的论述，现摘抄原文如下：

“自然科学不能同时在一切层次和细节上把握客观想象的规律性，而必须聚焦到一定的观测精度和描述水平。当我们写下质子和中子的符号 p 和 n 时，通常不必关心它们由那些夸克组成，而只需知道其电荷和质量。在化学中使用原子符号写出分子结构式时，H、C、O、N 等记号主要代表一定的原子量、离子半径、亲和力、化学价，而无须涉及原子核的构造。对于由几十个原子组成的嘌呤和嘧啶，化学结构式并非处处必需。人们引入 A、C、G、T 等等符号，更注意它们形成双螺旋时提供两个还是三个氢键。20 种符号所代表的不同氨基酸，更重要的不是“大同”的化学结构，而在于“小异”所导致的亲水、疏水、极性等种种性质的差别。

粗粒化不可避免地导致符号描述。在核酸和蛋白质的情形下，粗粒化都导致了一维的符

号序列。其实，对于有限长的符号序列，一维和高维并没有原则性的差异。高维序列可以归结为具有远邻关系的一维序列。这当然比只有最近邻关系的一维序列复杂些。但对于多数一维序列，也必须研究其中的长程关联。人们用符号序列来表示研究成果，更希望从符号序列中解读出自然界的规律[13]。”

具体说来，DNA 链是由 A、C、G、T 四个字符组成的一维序列，而多肽链是由 20 种常见氨基酸所对应的字符组成的一维序列。实际上，用字符 A 表示腺嘌呤这个物理实体，是隐藏了所有深层次的内部信息，而将所有的腺嘌呤视为全同粒子；与此同时，又绝对化了不同核苷酸之间的差异，即将腺嘌呤（A）与鸟嘌呤（G）、胞嘧啶（C）等其他的核苷酸，视为彼此没有共性的东西。因为，从符号上，我们看不出此 A 与彼 A 之间的差异，也看不出 A 与 C 的共同点。从信息来源的角度看，符号化后，DNA 链之间的差异，仅反映在字符的组成与排列上，没有其他的内在信息来源了。这是分子生物学中心法则带给我们的“相当粗糙”的物理模型。

1.3.2 生物序列分析

生物序列分析的基本思想是基于分子生物学中的一条经验规则，即当两个分子享有相似的序列时，由于进化关系或者物理化学限制，它们将很有可能具有相似的三维结构和生物学功能。因此，生物序列分析的首要任务是找出可以扩展到结构和功能性质的序列特征，从而阐明或长或短的字符序列的生物学意义。而序列分析的核心就是序列的比较，这包括同一序列内不同片段的比较，以及两个或多个序列之间的比较。比较的内容，从序列组分的变化，寻找特殊字段，到序列间字母的对应。比较的主要目的在于阐明序列之间的同源关系，以及从已知序列预测新序列的结构和功能。从处理对象上看，生物序列分析主要是核酸序列或蛋白序列的分析。

生物序列分析的基本问题有三个：（1）特定序列的寻找；（2）序列特征的提取；（3）序列分类。

（1）特定序列的寻找是指：

a. 寻找有特定生物学意义的 DNA 或多肽序列。如生物聚合酶的剪切位点(splice site)、翻译或转录的起始位点（initial site）、肽链上生物多聚酶的结合位点等。一般而言，在这些位点附近的序列保守性较高，否则，很难处理。

b. 寻找两条或（多条）序列的最大公共子序列，即序列的比对。

（2）序列特征的提取。如蛋白质编码区(protein coding sequence)的特征提取等。

（3）序列分类。根据序列统计特征上的差异，把生物序列归类。如将 DNA 序列分成编码区与非编码区，将多肽序列分成不同的二级结构或二级结构类等。

1.4 现有算法概述

1.4.1 海量数据期待有效算法

图 1-1 生物信息数据、Medline 生物医学文献、半导体芯片集成度的增长趋势

伴随着人类和其他模式生物基因组及蛋白质组计划的实施，生物信息数据量呈指数递增。图 1-1 是 Mark Bogulski 在 Bioinformatics 上发表的，虽有些老，但依然可以用来说明生物数据的增长趋势及作为人类知识积累的生物医学相关文献的增长情况，两者增长速度的差异加剧了数据增长与知识转化之间的矛盾。目前，生物信息数据库主要分成两个层次：一级数据库和二级数据库。一级数据库中存放的是直接来源于生物学实验的数据。这些数据只进行过粗略的整理，还没有经过分析、加工，称为 Raw Data，例如，基因组测序工程中产生

的DNA序列数据和各种图谱。二级数据库是由一级数据库以一定的规则派生出来的知识库，例如Motif库、蛋白质功能位点数据库Prosite、同源蛋白家族数据库Pfam。当前的一级数据库主要有三种类型的数据：DNA序列数据，氨基酸序列数据，蛋白质的三维结构数据。截止到2003年10月，已有大约200亿bp的DNA序列、13.5万条氨基酸序列和近2.3万个蛋白质三级结构的数据在网上公布。近几年来，又出现了与基因调控和代谢网络相关的数据库。面对如此巨量数据，单凭手工的方法来分析和处理它们，已难以胜任。于是，产生了一门以数学为基础，以计算机技术为手段，对生物数据进行分析和处理的一门新学科——生物信息学（计算生物学）。

生物信息学的主要任务就是从这些海量的已知生物数据中找出有用信息（data mining）从而把Raw data转化成Knowledge，进而揭示生命的奥秘。由于生物学一贯属于实验科学，其精确理论部分相对薄弱，目前的生物信息学，在一定程度上担当着理论生物学的角色，尽管有些时候，表现得很不称职[14]。目前，生物学的理论研究课题大体分为三个层次。第一，从DNA序列中识别编码蛋白质的基因及调控基因表达的各种信号，根据基因组的特征构建生物进化关系树等。其次，从氨基酸序列出发，根据已有的知识，建立蛋白质折叠过程的模型，预测蛋白质的结构和功能等。第三，根据已经积累的蛋白质个体的结构和功能知识，建立相互作用网络，重现生理过程。至于所采用的方法（论），可谓，八仙过海，各显神通。对于上面的第一层次，主要应用统计学和模式识别的方法。第二个层次，可以应用统计学方法，也可以应用物理（如分子动力学等）和化学（如量化计算等）的理论。而第三个层次，则需要综合运用系统论、控制论、信息论、耗散结构理论、协同学理论、超循环理论等横断学科和数学、物理学、化学、生物等基础学科的知识，并以工程学的方法组织研究。其中，对于以字符串形式出现的序列数据所常用的方法，将在下节中给出简要的描述。法无定法，万法归宗。不论用什么方法，都是为着一个目的：找出这些生物数据中所蕴含的规律。目前，已取得了一些重要发现，如DNA序列的蛋白质编码区有3周期性、DNA（或氨基酸）序列中存在长程关联（相互作用）等。但一些重要课题，如原核生物中短序列编码区的识别问题、真核生物中剪切位点的识别问题、基因调控区的识别、蛋白质形成的折叠过程（涉及到介观层次的动力学）、蛋白质二级及更高级结构的预测问题、蛋白质结构与功能关系、分子相互作用网络（代谢网络、基因调控网络等）的构建，生物整体生存态的序参数的寻找问题等，依然没有（很好地）解决。甚至对于上述问题提法的适定性，仍存在争议。还需要进一步认清问题的实质，发展更为有效的方法论。

鉴往知来，数据的大量积累往往预示着科学的重大发现即将出现。历史上，开普勒根据

第谷的天文观测数据，找出了行星运行的三大定律，为牛顿力学原理的产生做好了铺垫。世纪之交，生物医学领域同样出现了大量数据，于是，人们期待着类似的定律和原理的出现。目前，已有的发现，大多数属于统计规律，仍停留在“是什么”的层次，而对于内在机理，即解答“为什么”的问题，还所知甚少。这是可以理解的，知其然才能知其所以然，历史阶段往往是不容逾越的，因此，当前人们的主要任务还是解答“是什么”的问题。对生命现象的认识，仍需要长期的努力来逐步加深。

1.4.2 现有算法的粗略分类及简介

算法纷纭，名目繁多，这里只能是挂一漏万。每一种方法本身都是一门学问，在有限的篇幅内不可能面面俱到。很多方法都是点到为止，描述稍多的，也只侧重于交待基本思想或其内容要点。算法都具有一定的通用性，因此，按其所处理的问题或是应用的领域对其进行分类，未必是最合理的。但是，图 1-2 中的分类方法，反映了不同算法在特定问题的处理中出现的频繁程度。

序列比对 序列比对是真正把字符串当字符串来处理的方法。无论是核酸序列还是氨基酸序列，都可以用序列比对的方法加以处理。序列比对是寻找序列相似性的过程，前提是找到一种最佳的对齐方式，即使字符的匹配数目最大，或者是，使两条序列间相互转换时所需要的编辑操作：例如替换、插入或删除等的次数最少的对齐方式。这在具体实现时往往表现为，对给定的打分函数进行寻优。序列比对中，为了度量两条序列之间的相似性，有时要考虑序列中字符之间的替代关系，这通常由“突变矩阵”（Mutation Matrix）或叫“打分矩阵”给出。核酸序列比对时，通常不允许替换，即不使用替代矩阵（或说使用单位阵），BLASTN 程序也允许使用一些简单的替代关系。蛋白序列比对一般都会使用替代矩阵，有两个系列可供选择：PAM 系列[15]和 BLOSUM 系列[16]，而其中尤以 PAM250 和 BLOSUM62 最为常用。序列比对，按一次同时比对的序列数目分为：双序列比对(pairwise alignment)和多序列比对(multiple sequence alignment)；按比对的区域大小和优先次序，可以分为全局比对和局部比对。事实上，比对的概念还可以扩展到其他类型的数据库，例如用来比较蛋白质三维结构的三维结构比对，用来检测氨基酸序列与一个三维结构库的相容性的三维一维比对，以及用来比较两个生化途径的比对等。

图 1-2 生物序列分析方法的粗略分类

最早出现的一种用于双序列比对的方法是由 Gibbs 和 McIntyre 在 1970 年提出的一种可视化的方法——“Dot Matrix”[17]。该方法既可用于两条不同序列之间的比对也可用于一条序列与自身的比对，并以直观的形式（点的对角延伸）显示比对结果。后来，在 1981 年，Maizel 和 Lenk 发展了各种过滤和颜色显示表，大大增强了该方法的实用性。该方法目前还在序列分析中发挥着重要的作用[5]。自动化序列比对算法的理论基础是动态规划。动态规划是 R. Bellman 在 1950 年代发明的一种数学方法[18]。它把大问题按时间步骤或空间分布分割处理，从局部最优逐步寻求全局最优结果。这通常表示成一套逐点向前递推的关系，在完成向前推算后再回溯找到最佳路径。最早由 Needleman 在 1970 年提出的寻找氨基酸序列的最优整体比对的方法，实际上就用到了动态规划[19]。后来，Smith 和 Waterman 把动态规划算法用于寻求序列局部最优的比对[20]。虽然，动态规划方法利用递推减少了运算量，有

效地抑制了计算量的组合爆炸，但其计算量仍正比于 N^2，N 是问题规模的大小，比如序列长度。因此，在很长的时间里，人们无法使用此方法实现序列比对，而不得不满足于半经验的直观方法，诸如 FASTA 和 BLAST 所使用的方法。随着计算机速度的增长，这种限制已不严重。现在，越来越多的生物信息中心开始提供使用 Smith-Waterman 算法的服务[7]。

现在，流行的两两序列比对算法依然是 FASTA 和 BLAST，这两种算法都是近似算法。FASTA 的前身是 FASTP[21]，它的出现要早于 BLAST，但 BLAST 有后来居上的趋势。FASTA 是在 Dot Matrix 的基础上，通过散列法初步搜索候选的对角延伸，再在对角延伸周围限定一个区域，并在该区域内搜索最佳匹配，从而极大地缩短了动态规划需要的计算时间[22]。BLAST 是一种启发式的算法[23]。它首先寻找在两条序列中都出现的字母块，然后试图扩展匹配的区域，直至打分函数的分值不再增加为止。现在，FASTA 的版本为 3.0 版。BLAST 的当前版本为 2.0 版[24]。较之以前的版本，新版的 BLAST 增加了插入空位(Gaped)和位置特异的迭代功能(PSI-BLAST)。PSI-BLAST 的使用，有利于搜寻远亲同源关系。一般认为，BLAST 运行较快，对蛋白序列的搜寻更为有效；FASTA 运行较慢，对核酸序列更为敏感。比对的实现，除了高效的比对算法外，还需要一个高质量的数据库。常用的数据库为核酸和蛋白序列的无冗余库（non-redundant）。多序列比对由于其运算量大，实现起来比双序列比对要难，而且应用的规模也受到很大的限制。常用的多序列比对软件有 Clustal 系列[25]和近年来出现的 T-coffee[26]。为了确保序列比对找到的相似性有真正的生物学意义，还要用比对分数的统计学特性来帮着评估比对的可信性，即计算比对的统计置信度[27]。

另一个与序列比对紧密相关的领域就是亲缘树构建。在分子生物学层次上，亲缘树就是依据生物序列（通常为蛋白序列）的相似程度，反映不同物种间在进化上（或叫系统发育上）的远近亲疏的关系。常用的亲缘树构建方法有：“UPGMA”（Unweighted Pair Group Method with Arithmetic Means）方法和 NJ（Neighbor Jointing）方法[28]。序列比对给出的相似性大小的结果，可以直接作为聚类时距离的度量，并借助于一些计算机程序（例如 PHYLIP、Treeview 等）直接显示出序列间的亲缘树。近年来，出现的 COG（Clusters of Orthologous Groups of proteins）数据库就是基于一种基因组范围内的蛋白序列间的“多对多”的双向序列比对而构建的，并已在基因组进化和基因功能注释中得到了应用[29,30]。总之，序列比对构成了生物序列分析方法的一个大分支，是生物序列分析中的基本方法。

核酸序列的分析方法

马尔科夫模型、隐马尔科夫模型及相应的 EM 算法 马尔科夫模型，又称马尔科夫链，

在 Markov 假设的基础上，可以用来描述离散随机过程。模型的参数是初始状态的概率（用向量表示）和状态转移的概率（用矩阵表示），其数目由状态的个数和模型的阶数共同决定。对于 DNA 序列来说，一阶马尔科夫链有 4 个初始概率和 16 个状态转移概率，归一化后，共有 15（=3+12）个独立参数。计算 3 核苷酸频度的 2 阶马尔科夫链共有 15+48=63 个独立的参数。一般地，计算 *K* 长 DNA 字符串的（*K*-1）阶马尔科夫模型的状态初始概率和转移矩阵，总共确定 4^{K+1}-1 个独立参数。目前，许多寻找基因的程序中使用的 5 阶马尔科夫链对应于 K=6，共有 16385 个参数，主要刻画 DNA 序列中六核苷酸的统计特征。以上给出的参数计算公式，是针对齐次马尔科夫模型的，若模型为非齐次，即转移矩阵与位置有关，则参数的数目还会增加[7]。一般说来，参数越多，需要的训练集就越大。齐次马尔科夫模型常用于描述非编码区，周期 3 的非齐次模型则被用来描述编码区[31]。

隐马尔科夫模型，是在马尔科夫模型的基础上，增加了输出概率，即在某一特定状态下，以一定的概率输出一个值。输出的值也称为观察值，相对地，把产生观察值的状态，称作隐含状态(hidden states)。在生物序列分析中，观察值的集合也是离散的。隐含状态到输出值的概率用输出概率矩阵（confuse matrix）来表示。若隐含状态数为 n，观察值的数目为 m，则输出概率矩阵的大小为 n×m。隐马尔科夫模型有三个基本问题，即评估问题、解码问题和学习问题，分别用向前算法、韦特比(Viterbi)算法和向前向后算法来解决。在基因识别中，一般选取编码、非编码、编码之补等状态作为隐含状态，而观察值就是四种核苷酸 A、C、G、T[32]。而从 DNA 序列中识别出编码区的问题，就是一个解码问题，用韦特比算法求解。因为，韦特比算法也是基于动态规划的，因此，运算量为 $O(n^2)$。近年来，马尔科夫模型或隐马尔科夫模型已成为基因识别领域的主流算法，著名的基因识别软件 Glimmer 系列[33-35]和 GenMark 系列[31,32,36]都是基于马尔科夫模型或隐马尔科夫模型的。

EM 算法，是求解最大似然估算问题（Maximum-likelihood estimation problem）的一般性方法。用于隐马尔科夫模型学习问题的向前向后算法，就是 EM 算法的一个特例。EM 算法通常分成两步，即所谓的 E-step 和 M-step。E-step 是计算给定的参数（分布）下的期望值；M-step 则须求出使期望值最大的那个参数值（或参数的修正值），进而修正原参数。如有必要，反复执行 E-step 和 M-step，直至期望值达到最大（至少是局部极大），此时的参数即为最优参数。用于基因识别的隐马尔科夫模型如果采用 EM 算法学习，可以自恰地给出模型的全部参数，而不需要已知的注释基因作为其训练集[37]。EM 算法还曾被用来识别蛋白质绑定位点[38]。

傅立叶变换（或功率谱）与小波变换 傅立叶变换来源于傅立叶积分，是频谱分析的常用工具。如果把核酸序列看成随时间分布的一维信号，就可以应用傅立叶变换，观察其在频域上各频率的组分，因而可以发现序列中蕴含的周期性。应用傅立叶变换于 DNA 序列分析的论文有[39-41]。小波变换是傅立叶变换的重要发展。不同于傅立叶变换的、具有无穷支集的周期基函数，小波变换选用具有紧支集的小波函数作为基函数，因此小波变换可以检测信号的局部突变。在结合了多分辨分析之后，类似于视觉过程的由远而近，小波变换可以给出一个从粗到细（coarse-to-fine）的、多尺度的观察序列，而 Mallat 算法（亦称快速小波变换算法）则保证了信号分解与重构的实现[42]。小波变换在 DNA 序列分析[43]中，可以用来检测具有不均匀组分的区域及确定其边界，例如，寻找致病岛（pathogenicity islands）[44]、外来基因、或检测 Isochore 的边界[45]等，而尤以 Haar 小波较为实用。小波在生物信息学中应用的综述见[46]。

几何表示方法 以字符形式出现的 DNA 序列，适合于在计算机中存储与处理，却不适合给人看。若能将其转化成某种几何形式，特别是，若能转化成一条曲线，就能很直观地进行观察与比较。基于这一想法，Hamori 和 Ruskin 在 1983 年，提出了 DNA 序列的三维曲线表示，即所谓 **H curves**，并指出该表示方法适合于长序列的观察比较，但未证明这种表示方式与原 DNA 序列的等价性（实际上是等价的）。后来，张春霆等人基于核苷酸对称性的观察，提出了 **Z 曲线理论**[48,49]。张春霆等人建议用正六面体来表达碱基之间的对称性关系，并指出在正六面体的一个内接正四面体内存在一点，该点的坐标恰好构成四种核苷酸频率归一化后三个独立参数的一种表示方式。当沿着 DNA 序列从头走到尾时，可以得到一系列这样的点，将相邻两点用适当的曲线连接后所得到的曲线，就称为**表示 DNA 序列的 Z 曲线**。作者证明了 Z 曲线与其所表示的 DNA 序列之间存在一一对应关系，既 DNA 序列的 Z 曲线表示是一种等价表示。Z 曲线理论近年来在 DNA 序列分析中得到广泛的应用[50,51]，特别是在基因识别领域，随着 Zcurve 系列软件[52,53]的推出，其影响越来越大，受到学术界越来越多的关注。

语言学方法 自从提出“中心法则”以来，转录、翻译、编辑、修饰等具有语言学背景的术语就在分子生物学中广泛应用。由于生物遗传语言与人类自然语言有许多相似之处，经过一定的抽象后，语言学（linguistic 而非 philology）的方法可以在生物序列分析中发挥作用[7]。当前，生物信息学中所使用的语言学方法主要是指“形式语法”。形式语法源自自然语言的数学模型，与自动机理论有对应关系，例如，在 Chomsky 体系中，正规语法对应于

有限自动机，上下文无关语法对应于下推自动机，上下文有关语法对应于线性有界自动机，而无约束性语法则对应于图灵机[54]。形式语法用于模式识别问题，形成了**结构模式识别**方法。结构模式识别方法又叫**句法结构模式识别**，它只用到了形式语法的一部分理论，同时，为了适应模式识别问题的需要，又对“形式语法”理论进行了扩展，例如，发展了前后文无关程序语法、高维模式语法、PDL 语法、树语法等。相应于统计模式识别方法中的基于样本的算法学习，句法结构模式识别中有基元选择和语法推断，而预测过程在句法结构模式识别中则表现为句法分析。句法结构模式识别的优势在于，它不仅可以确定模式所属的类别，还可以给出模式的结构信息[55]。1994 年，Searls 等开发了基于语言学方法的基因结构识别程序 GeneLang[56]。1996 年，Pesole 等人则用语言学方法分析了核酸序列的编码策略[57]。关于语言学方法在生物序列分析中的运用，可参看 Searls 的综述[58]。

其他方法

决策树 简单地说，决策树就是一系列需要回答“是”或“否”的问题，按一定的层次结构组织起来，最终导致一个确定的结论（决策）[59]。决策树学习是以实例为基础的归纳学习算法。它着眼于从一组无次序、无规则的事例中推理出决策树形式表示的分类规则。它采用自顶向下的递归方式，在决策树的内部结点进行属性值的比较，并根据不同属性值判断从该结点向下的分支，在决策树的叶结点得到结论。决策树学习算法的一个最大的优点就是他在学习过程中，不需要使用者了解许多背景知识（这同时也是它最大的缺点），只要训练例子能够用**属性—结论**式的方式表达出来，就能使用该算法来学习。决策树学习与很多统计学习方式等价，比如，前向人工神经网络的学习，支持向量机的学习等。早期的决策树学习算法是 1966 年由 Hunt 等人提出的 CLS 算法。它是单一算符的递归算法，其主要思想是从一棵空的决策树出发，通过添加新的判定结点来改善原来的决策树，直至该决策树能够正确地将训练实例分类为止。后来，Quinlan 于 1979 年基于信息熵下降，提出了 CLS 算法的改进形式：ID3 算法。ID3 算法还引入了增量式学习机制，扩充了决策树的学习能力[60]。在生物信息学中，决策树被用于基因识别[61,62]或蛋白质结构预测[63]。

后缀树 寻求两个符号序列的最长公共子序列的问题，早就出现在理论计算机科学中。它很容易同序列比对等实际问题联系起来。未经推敲的简单考虑，导致比例于 N^3 的算法，N 是问题的规模[7]。关于后缀树方法的详细描述可以参看 Dan Gusfield 的著作[64]。

近年来，后缀树方法被用来进行基因组水平的序列比对[65,66]。后缀树方法用于序列比对时，大体分三步：（1）识别所有的 MUMs（Maximal Unique Matches），所谓“最大唯一

匹配”，就是指两条序列的最长公共子序列；（2）对 MUMs 进行排序。（3）处理四种空位——重复、SNPs、插入和可变多态性区域。后缀树比对方法的计算时间与数据量和序列之间的相似程度都有关系：所比较的基因组的长度越长相似性越差，则计算得到第一步和第二步结果所用的时间越长，反之亦反；所比较基因组的空位越多则得到第三步结果的时间越长，反之亦反。

另外，后缀树作为一种通用的数据结构，可用于核酸或氨基酸序列的计算机存储与自动化处理，在提取 DNA 序列中的结构模体[67]和蛋白质序列数据库的分类中得到了应用[68]。

判别分析 判别分析是用于判别个体所属群体的一种统计方法。它的特点是根据已掌握的、历史上每个类别的若干样本的数据信息，总结出客观事物分类的规律性，建立判别公式和判别准则。然后，当遇到新样本时，只要根据总结出来的判别公式和判别准则，就能判别该样本所属的类别[69]。只依赖于样本信息的基本判别方法为距离判别法（马氏距离、欧氏距离等）。若还依赖于先验信息，则有贝叶斯判别方法。判别分析作为一个强有力的统计模式识别方法在 DNA 序列的模体寻找中得到应用。依据样本类边界的形状，判别分析简单地分为线性判别和非线性判别。线性判别如 Fisher 判别，应用的最多，而在非线性判别中，二次判别曾被用来识别人类基因组中的编码区[70]。关于判别分析在 DNA 序列分析中的应用，可以参看张奇伟的综述[71]。

位置权重矩阵（数组） 位置权重矩阵是一种序列模型，它反映了不同位置处各字符出现的频率，适用于以保守序列为特征的功能位点的识别。而位置的表示需要选取一个参考点。通常的情况是，选取待识别的位点作为参考点，位置在其上游为负，下游为正。较早的用于信号序列剪接位点识别的文献见[72]。后来，张奇伟等人发展了权重数组方法，用于剪切信号分析[73]。

分形特征 我国学者郝柏林等较系统地研究了基因组的分形特征[74]。他们建议用单核苷酸方阵通过作矩阵的直乘来表示 K 长 DNA 串的频度矩阵。该矩阵的每一个元素表示 K 长 DNA 串的一种组合方式，该组合方式在基因组中出现的频度用矩阵元素值的大小来表示。把频度矩阵摆放在屏幕上，依据矩阵元素值的大小给屏幕的各点上色，就得到基因组中 K 长 DNA 串出现频度的彩图[75]。据此彩图便可以研究基因组的分形特征。关于 DNA 序列的分形特征，最近还有讨论[76]。

混沌游戏表示 混沌游戏表示是 Jeffrey 在 1990 年提出的，用于表示基因结构[77]。1994

年，Fiser 等人又用它来表示蛋白质的结构[78]。用该方法于基因组序列分析的文章见 Almeida 的[79]。后来，Almeida 等人又进一步发展了该方法，将其推广到一般形式，提出了所谓“通用序列映射”（Universal Sequence Map）[80]。

阶乘矩 阶乘矩本是高能粒子物理中研究多粒子产生机制的数学工具。2000 年，Mohanty 等人把它用于 DNA 序列的研究[81]，宣称找到了 DNA 序列的一个特征长度范围（characteristic length scale），但很快就有人对此提出商榷[82]。

蛋白质二级结构预测与机器学习

前面提到，蛋白质二级结构是“蛋白质二级结构辞典”中定义的蛋白质分子的几种主链结构类型。蛋白质二级结构预测就是从蛋白质的一级序列出发，预测序列中各分子所属的二级结构类型。抽象出来看，就是从 20 种氨基酸组成的序列到 3 种二级结构类型（3 类预测）或 8 种类型（8 类预测）组成的序列的一个映射。预测结果的好坏就是看，谁构造的映射精确，并且泛化能力强。迄今，蛋白质二级结构预测算法共经历了三代[83]。第一代是指上世纪六七十年代的工作[84-87]，这些算法几乎全部都是基于单个氨基酸倾向性的。第二代算法大体是指上世纪九十年代之前的算法，此阶段的算法主要考虑的是 3-51 个相邻残基片段的倾向性，三类预测的准确率在 60%多，此时已开始使用机器学习类算法[88]。第三代预测算法是指上世纪九十年代之后的算法，此时蛋白质二级结构预测领域已经是机器学习类算法特别是人工神经网络的天下。这一代算法除了考虑残基片段的局部信息以外，还把从序列比对得到的进化信息（全局信息）结合进来[89]，把 3 类预测的准确率提高到 70%以上[90]。这些算法通常的做法是，把待预测的序列拿去和蛋白质序列的无冗余库（nr）作比对，并以比对结果所给出的概貌（Profile）作为神经网络的输入，再由多层神经网络预测二级结构。不久前，有人宣称 800 多个神经网络联合使用，把二级结构 3 类预测的准确率提高到 80%左右[91]。最近又出现了用回归神经网络预测蛋白质 8 类二级结构的做法[92]。关于蛋白质二级结构预测的综述，较早的可以参看[93]，近来的有[94,83]。蛋白质二级结构预测常用的训练和评价数据库有 RS126[90]、CB396 和它们的混合 CB513 等[95]。最常用的蛋白质二级结构预测的评价指标是 $Q3$，它就是 3 类预测中预测对的氨基酸的数目与氨基酸总数的比值。后来 Rost 等人考虑到蛋白质 3 维结构的变动性，在 1994 年重新定义了一个评价指标，叫做“Sov”（segment-overlap）[96]。1999 年，又有人改进了它的定义[97]。最近，我国学者张春霆等人针对蛋白质二级结构预测，又提出了具有“含量平衡能力”的评价指标，称为 $Q9$，丰富了预测评价问题的研究内容[98,99]。

还存在与蛋白质二级结构预测相关的课题，叫“蛋白质二级结构类”预测，它是从整体上预测一段氨基酸序列所形成的二级结构类别。通常定义了 4 种二级结构类别：全α型、全β型、α/β型、$\alpha+\beta$型。我国学者张春霆等人采用 20 维氨基酸组成空间中向量表示的方法，在蛋白质二级结构类预测方面，取得了引人瞩目的成就[100]。

下面介绍几种蛋白质二级结构预测领域中常用的方法。

人工神经网络 人工神经网络是对生物神经网络信息处理过程的极其粗浅的模拟。它设想，具有简单的信息处理功能的神经元（细胞），依照一定的拓扑结构彼此互联，形成网络。信息就在此网络上，输入、变换、输出，从而完成信息加工的过程。人工神经网络，不仅具有仿生学意义，从而有助于人脑信息处理机制的研究，而且它还以其强大的数据拟合和优化运算能力，在工程领域得到广泛的应用[101]。其中，以误差反传算法（EBP，Error Back-Propagation）为其学习算法的前向网络是拟合型网络，由“完全性定理”保证的强大的非线性逼近能力，在蛋白质二级结构预测中得到了广泛的应用[83]。当前，蛋白质二级结构预测领域，称得上 state-of-the-art 级的预测软件，绝大多数都是基于人工神经网络的，例如 PHD[90]，NNSSP[102]，PSIPRED[103]等，所不同的只是在具体实现时，变换花样。运算型的 Hopfield 网络，结合模拟退火算法，以其较为强大的全局优化能力，可以在 RNA 或蛋白质三维结构预测中得到应用。具有自动分类能力的神经网络——自组织特征映射（Self-Organizing Feature Map）则被用来观察人类蛋白质序列的聚类[104]。

支持向量机 支持向量机，又叫内核机（Kernel Machine），是建立在统计学习理论的结构风险最小化原则之上的一种学习机器。针对两类分类问题，支持向量机实现的是如下思想：通过事先选择的非线性映射将输入向量映射到一个高维特征空间，并在该空间中构造最优分类超平面[105]。特征空间的维数高于测量空间的维数，是支持向量机的一个特色。为了处理高维空间中的内积运算，基于泛函分析中的一个结论，将高维空间中的内积转化为原空间中的一个函数。该函数就称为支持向量机的核函数，又叫作内积的回旋（convolution of the inner product）。所谓支持向量，是指处在两类分界面附近的、对分类起决定作用的样本点。支持向量机在支持向量上展开解，因此，其结构的复杂程度取决于支持向量的数目，而不是特征空间的维数[106]。又由于非线性映射的引入，可以将低维线性不可分转化为高维线性可分。

支持向量机作为一种新出现的“有导师的”（或叫受监督的，supervised）学习机器，近

年来，在生物信息学领域中得到了越来越多的运用。除了用于蛋白质二级结构预测[107]外，还被广泛运用于蛋白质功能分类[108,109]、生物医学文献的组织与信息挖掘[110]（为了研究蛋白质相互作用）、与药物抗癌机理相关的基因的识别[111]、翻译起始位点[112]和剪切位点的识别[113]、癌组织样本基因表达数据的分类和确认[114]等。

多元回归 回归分析研究一个（或多个）因变量与一个或多个解释变量之间的相互依存关系，并估计或预测在解释变量的数值已知或固定的基础上因变量的平均值。多个解释变量的，叫多元回归；多个因变量的，叫多重回归。由于因变量被定义成随机变量，具有一定的概率分布，解释变量则既可以是普通变量也可以是随机变量，这种相互依存关系是一种不同于确定性函数关系的统计关系[69]。回归分析要解决的问题，一是根据试验或观测数据选定适当的回归函数，或检验某种选定的回归函数是否合用。二是基于若干观测数据对回归函数中的未知参数进行估计。三是检验有关这些参数的假设。四是对随机误差的影响程度进行估计，最常用的是随机误差的方差估计。五是利用已建立的回归方程进行预测和控制[115]。

在多元回归分析中最常用的是线性回归。这里的“线性”有两层涵义：一是因变量对解释变量存在线性函数关系；二是因变量对未知参数存在线性函数关系。线性回归通常是第二种涵义。最常用的参数估计方法是最小二乘法。在随机误差满足正态分布的前提下，参数的最小二乘估计量在线性无偏估计量中，具有最小的方差。

在生物信息学领域，较早的时候，多元回归作为定量化的分析手段，被用于核酸序列与功能活性之间的关系[116]以及大肠杆菌 DNA 序列中核糖体绑定位点的研究[117]。后来，又被用来进行蛋白质二级结构含量[118]和蛋白质二级结构的预测[119]。近来，多元回归分析又在基因芯片表达数据的聚类边界的统计估计中得到应用[120]。

关于生物序列分析领域中常用算法的介绍，可以参看教材[121]和[122]，也可参看工具书[7]和该领域的综述与展望之一[6]。

1.4.3 现有算法的特点及存在的问题

上述算法，各有特点，很难用一两句话概括。除了序列比对方法外，其它各方法多数都是从其他领域中借用来的，并非专门针对字符处理的。下面在很粗的视角下，谈一下现有算法的特点及存在的问题。

（1）随机性。现有的序列分析方法，主要是统计模式识别类方法。统计方法，是基于概率论的，不确定性便是它的一个显著特点。如今在 DNA 序列分析中，居主流地位的 Markov

模型体系，是把生物序列当随机过程看待的，是字符序列的随机过程模型。生物序列虽然偏离随机序列不远，但并不是真正的随机序列，特别是，蛋白质编码区有着明显不同于随机序列的特征（如 3 周期性），因此，并不是所有的生物序列都适合用随机过程描述。此外，当前的一些以随机性为其特征的算法是无法得到确定性结论的，它所给出的结果往往是一个概率分布。（2）现有算法的知识表达能力较弱。上面提到，当前生物信息学，以生物序列分析为其主要内容，致力于解决“是什么”的问题。换言之，字符序列的哪些排列组合方式是编码区，哪些排列组合方式是非编码区，哪些排列组合方式形成螺旋，而又是哪些排列组合方式形成折叠片。上述问题，本质上是一个排列组合问题，**组合数学**的方法有可能给出简明而确切的答案。而当前，在涉及氨基酸序列的问题中，如蛋白质二级结构预测、二级结构含量的预测以及序列与功能活性之间关系的研究等，行之有效而广泛运用的方法是机器学习（统计学习），像人工神经网络、支持向量机等。这些机器学习后获得的知识，只有那个学习机器（Learning Machine）知道，而人并不知道。而且，这些知识以大量参数的形式存在（如，人工神经网络的连接权矩阵），很难从中提炼出简单明白的指导原则，丰富人类认识自然的知识宝库。

尽管当前的算法存在着上述的特点和问题，但它们的运用，仍是取得了良好的效果。例如，目前，原核生物中的基因识别程序，其准确率已经超过了 90%，而在蛋白质二级结构预测领域，三类预测的效果也已达到 80%左右。前面提到，序列分析的核心是序列的比较。除了序列比对是直接比较字符串以外，其它的统计类方法，进行序列比较的前提都是字符序列的定量化表示。不论采用什么量化方法，最终都要把序列变成一个数（实数，或叫标量），因为只有实数才可以直接比较大小。既然如此，那么，一个自然的想法就是，何不把字符序列直接对应成一个数？而另一方面，字符序列分析的实质是排列组合问题，与排列组合密切相关的一个学科就是**数论**。因此，本文尝试着把字符序列看成是数的表示，从而把字符序列分析问题，转化成一个数论问题，进而求解。

1.5 从数论的角度看，生物序列分析的三个基本问题

数论是一门很古老的学科，同时，它也是一门长久不衰、十分活跃的学科。数论研究的是数的性质，特别是整数的性质。历史上，一些数论中的著名问题，如素数分布问题、歌德巴赫猜想、华林问题等，吸引了大批的大数学家的研究兴趣[123]。我国学者华罗庚、陈景润、王元和潘承洞等，都是数论方面的专家，他们的研究在国际上处于领先地位。数论依据

所使用的研究方法，分成初等数论、解析数论和代数数论等[115]。数论研究饶有兴味，却也难度很大，需要一定的天分。下面从数论的角度看一下，生物序列分析的三个基本问题。

（1）特定序列的寻找

a. **全嵌入子序列** 设有 m 个字符组成的字符集 C，s_1 和 s_2 是由 C 中的字符所构成的字符序列，它们分别表示整数 N 和 n，其中 N 的位数为 L，n 的位数为 l（即 s_1, s_2 的长度分别为 L, l，$l \le L$）。若 s_2 完全嵌入 s_1 中（即 s_2 为 s_1 的子序列），设 s_2 的首字符在 s_1 中出现的位置为 $l_1 (0 \le l_1 \le L - l)$，则有下式成立

$$n = \frac{N \bmod m^{L-l_1+1} - N \bmod m^{L-l_1-l+1}}{m^{L-l_1-l+1}} \tag{1-1}$$

这就是全嵌入子序列所满足的关系式。在数论中，上式表示同余关系 $N \bmod m^{L-l_1+1} \equiv N \bmod m^{L-l_1-l+1} (m^{L-l_1-l+1})$，即，$N \bmod m^{L-l_1+1}$ 同余于 $N \bmod m^{L-l_1-l+1}$ 模 m^{L-l_1-l+1}。从该方程中解出 l_1，即找到了 s_2 出现在 s_1 中的位置。

b. **两序列的公共子序列** 设 s_1 和 s_2 的公共子序列为 s。s_1、s_2 和 s 分别表示整数 N_1、N_2 和 n，它们的长度，即 N_1、N_2 和 n 的位数分别为 L_1、L_2 和 l。并设 s 的首字符在 s_1 中的位置是 $l_1 (0 \le l_1 \le L_1 - l)$，在 s_2 中的位置是 $l_2 (0 \le l_2 \le L_2 - l)$。则：

因为 s 是 s_1 的子序列，故有

$$n = \frac{N_1 \bmod m^{L_1-l_1+1} - N_1 \bmod m^{L_1-l_1-l+1}}{m^{L_1-l_1-l+1}}$$

因为 s 是 s_2 的子序列，故有

$$n = \frac{N_2 \bmod m^{L_2-l_2+1} - N_2 \bmod m^{L_2-l_2-l+1}}{m^{L_2-l_2-l+1}}$$

又由于 s 是 s_1 和 s_2 的公共子序列，于是有

$$\frac{N_1 \bmod m^{L_1-l_1+1} - N_1 \bmod m^{L_1-l_1-l+1}}{m^{L_1-l_1-l+1}} = n = \frac{N_2 \bmod m^{L_2-l_2+1} - N_2 \bmod m^{L_2-l_2-l+1}}{m^{L_2-l_2-l+1}}$$

即，

$$\frac{N_1 \bmod m^{L_1-l_1+1} - N_1 \bmod m^{L_1-l_1-l+1}}{m^{L_1-l_1}} = \frac{N_2 \bmod m^{L_2-l_2+1} - N_2 \bmod m^{L_2-l_2-l+1}}{m^{L_2-l_2}} \tag{1-2}$$

上式称为“序列比对的第一数论方程”。它表示字符间无替代关系时，l 长公共子序列在两字符序列中的位置与两序列所表示的整数之间的关系。它是一个含有三个未知数 l, l_1, l_2 的不定方程，其中，$0 \le l \le L_1, L_2$，$0 \le l_1, l_2 \le L_1 - l, L_2 - l$。该方程的解是三元组 $\{l, l_1, l_2\}$，设其解集为 Ω，其中，相应于 l 最大的解为 $\omega^* = \{l^*, l_1^*, l_2^*\}$，则 s_1 和 s_2 的相似性度量可定义成

$$r = \frac{2l^*}{L_1 + L_2}, \tag{1-3}$$

$0 \le r \le 1$，即 r 的取值范围介于 0 和 1 之间。当且仅当，两字符序列 s_1 和 s_2 没有公共子序列时，$l^* = 0$，$r = 0$；当且仅当，两字符序列 s_1 和 s_2 全同时，$l^* = L_1 = L_2$，$r = 1$。

因此，两序列比对的过程，可以看作是求解不定方程（1-2），并在解空间中找出 l 最大的解 ω^* 的过程，

$$\begin{aligned} &\omega^* = \{l^*, l_1^*, l_2^*\} \\ &l^* = \max_{\omega \in \Omega}(l) \end{aligned} \tag{1-4}$$

一旦找到包含 l^* 的解 ω^*，将其代入(1-1)式，就可以由 s_1 或 s_2 求出最大公共子序列 s^*。

上面，“序列比对的第一数论方程”是根据同余关系给出的，而同余关系，从字符序列的角度看，在特定的情形下，是指后缀相同。因此，上面基于数论的序列比对方法，在实现上，与后缀树算法等效，它们的研究可以相互促进。

（2）特征提取问题 特征提取是从测量空间到特征空间的一种变换，通常是降维变换。特征提取的目的是为了便于分类。若从数论的角度看，特征提取问题是一个用较小的数表示较大的数的问题，或者说是一个用“概数”表示“确数”的问题，而这种表示又能使同类的数彼此靠近，不同类的数彼此远离。一个较容易想到的方法，就是通过分解质因数，将所研究的大数在素数轴上展开。但该方法可能是行不通的，因为，两个仅相差 1 的数，可能一个是合数，一个是素数，分解后的结果相差很大，从而不能表现两条序列的高度相似性。如果注意到自然计数制给出的**数系**，是由两部分组成的，即**基数**和**位权函数**，而**一个数就是以位权函数为权重的、基数的加权和**，就会想到，能不能在基数和位权函数上做点文章，从而解决序列的特征提取问题呢？这就是本文第二章所给出的“字符序列的解析数论模型”或

叫“对偶描述子方法”，所讨论的主要问题，也是该论文所讨论的主要问题。它的基本思想是把基数和位权函数的概念，从整数域推广到实数域（并进一步推广到 n 维实空间），从而把函数逼近理论与分析方法引入问题的求解。不得不说，这是一种退而求其次的方法，丧失了原本数论求解的精确性，在本质上，仍是一种统计方法。尽管如此，作为一条新的思路，该方法还是有其激动人心和引人入胜之处的。

（3）数字的分类 奇数、偶数，素数、合数，费马数、梅森数等等不同类型的数字有不同的特点。设想不同类的字符串，对应于不同类的数字，而它们是不同的**报数器**所报出的。那么，数字的分类问题就变成了**报数器的归属问题**。每一个报数器都有其内在的机理，该机理决定了报哪些数。比如，梅森数就是梅森报数器 2^n-1 所报出的数字。不同报数器所报出的数的分布，可能彼此之间有一定的重合。报数器的归属问题就是确定某一类数字是由哪个报数器报出的，最好是确定报数器的内在机理，比如，给出类似梅森公式的函数关系来描述同一类数。这个问题，也有相当的难度，就像要找到所有素数都满足的生成公式一样困难。这个问题等效于给定一个形式语言，找到能接受它的自动机，因此，数字分类的研究和形式语法的研究，是可以相互促进的。

从数论的角度看字符序列分析，只是**模式识别问题的数论方法**的一个特例。此前的模式识别方法，大体上分成两类：统计决策模式识别和句法结构模式识别。在统计决策类模式识别方法中，基于概率论知识，模式被表示成一个向量，而在句法结构模式识别方法中，基于形式语言理论，模式被表示成一个句子。本文提出的模式识别的数论方法，则把模式表示成一个数字，这是模式识别领域内的一个新思路。然而，囿于作者的知识范围和业务能力，对于上述问题，只能提供一点粗浅的思路。问题的真正解决，甚至确定问题能否解决，都要由未来的学术精英来完成了。这里只要能收到一点抛砖引玉之效，就足感快慰了。本文第二章尝试着给出，字符序列的“解析数论”模型，或者说，字符序列数字化方法的一般性数学描述，旨在从数论的角度出发，解决字符序列的特征提取问题，但所举例子，多来源于生物信息学领域。

... But I am very much excited by your article in May 30th [1953] Nature, and think that bring Biology over into the group of "exact" sciences. I plan to be in England through most of September, and hope to have a chance to talk to you about all that, but I would like to ask a few questions now. If your point of view is correct [,] each organism will be characteristized by a long number written in quadrucal (?) system with figures 1, 2, 3, 4 standing for different bases ... This would open a very exciting possibility of theoretical research based on combinatorix [sic] and the theory of numbers! ... I have a feeling this can be done. What do you think?

George Gamow (1904-1968), in a letter to Watson and Crick (1953).

第二章 字符序列的解析数论模型

最早在 1953 年，乔治·加莫夫(1904-1968)在一封写给沃森和克里克的信中提到，由于分子生物学中心法则的确立，未来的理论生物学研究将会是一个排列组合理论和数论的用武之地。然而，加莫夫先生在 1968 年就去世了，他没能赶上大规模的生物测序。后来，随着大规模生物测序工作的进行，把生物序列当作随机过程看待的统计类模式识别方法，逐渐成为生物序列分析问题的主流方法。这是加莫夫先生所始料未及的，却有着它的必然性，因为，组合数学和数论对于一般人来说太难了。首先接触到生物序列的是那些完成测序的工程技术人员，他们不大可能精通组合数学和数论，但是，统计类方法却是他们的拿手好戏，这也许就是生物序列的随机过程模型成为主流的重要原因之一。后来，在 1999 年，法国人 Delamarche 等用统计 4 字符排列组合的方法来寻找翻译起始位点[124]，但他们并未意识到数论方法在字符序列中的应用。作者于 2002 年春，基于字符串信息来源的分析，仿照自然计数制，把扰动法的思想引入统计领域，提出了字符序列数字化的方法，当时称为自然变量法。那时，作者还不知道加莫夫先生的思想，也不知道 Delamarche 等人的工作。后来，自然变量法曾易名为数字变量法，并在导师的指导下发展成现在的**解析数论模型**（Analytic Number Theory Model）。从数论的角度看待字符序列分析问题，即把字符序列看成是数的表示，从而把字符序列分析问题转化成一个数论问题的思想方法，叫做字符序列分析的数论方法。用分析方法来解决数论问题的方法叫做解析数论方法。本文所给出的字符序列分析的解析数论模型，专指把数系中的基数和位权函数的概念从整数域推广到实数域的做法，其核心概念是“对偶描述子”(Dual Descriptor)，因此，有时也称该模型为**对偶描述子方法**。

2.1 字符序列的数字化表示

2.1.1 字符序列的量化

量化被研究对象是科学研究的第一步。现在，我们的研究对象是长短不一的多字符序列。如何将它们量化呢？自然计数制基本上解决了这一问题。

设有 n 字符构成的字符集 $C=\{c_1,c_2,\cdots,c_i,\cdots,c_n\}$，用 C^* 表示由 C 中的字符组成的、长度有限的所有字符串（字符序列）所构成的集合。对 C 中的字符规定一个顺序，则序数构成的集合为 $N=\{1,2,\cdots,i,\cdots,n\}$。构造映射 $C\to N$，这种映射共有 $n!$ 种方式。要达到量化字符串的目的，任取其中一种即可。不妨就按照集合所列的顺序，取如下这种：

$$f_1:C\to N=f_1:c_i\to i=\begin{matrix}\{\ c_1\ c_2\cdots\ c_i\cdots\ c_n\ \}\\ \updownarrow\ \updownarrow\cdots\updownarrow\cdots\updownarrow\\ \{\ 1\ \ 2\cdots\ \ i\cdots\ \ n\ \}\end{matrix}。\qquad(2\text{-}1)$$

在映射 f_1 下，对每一个字符串 $s\in C^*$，定义数 N 与它对应。当 $s=\varepsilon$（空串）时，$N=0$；当 $s=c_{i_k}c_{i_{k-1}}\cdots c_{i_0}$ 时，

$$N=i_k\cdot n^k+i_{k-1}\cdot n^{k-1}+\cdots+i_1\cdot n+i_0 \qquad(2\text{-}2)$$

其中，$1\le i_j\le n, j=0,1,\cdots,k$。字符串 s 称作数 N 的以 n 为底的表示或 n 进制表示[125]。

如果按照习惯顺序从左向右数，设字符串 s 的长度为 L，则(2-2)可以写成：

$$N=\sum_{k=1}^{L}n^{L-k}\cdot i_{c_k} \qquad(2\text{-}3)$$

其中，i_{c_k} 表示出现在字符串 s 中的位置 k 处的字符 c_k 所对应的数字 i。

这样一来，任给一个 $s\in C^*$，都有一个非负整数 N 与它对应，并且这种对应是唯一的，即，通过 f_1 建立了 $C^*\leftrightarrow Z^+$ 的一对一映射，其中，Z^+ 表示非负整数集。

2.1.2 字符串的信息来源与加权统计

考察字符串的信息来源，我们发现，字符串所能提供的信息不外乎来自两个方面：组成和排列。排列方面的信息远比组成方面的信息丰富，因为，组成确定的字符串，其排列方式

还可以有很多种，而排列确定的字符串，其组成就完全确定了。

组成方面的信息可以由通常的**频率**去反映，看下面这个例子。

s_1	A	G	C	A	T	A	G	G	T	C	C	A	C	A	G	T	T	G
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18’

s_2	A	G	C	A	T	A	G	G	C	T	C	A	C	A	G	T	T	G
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18.

s_1 和 s_2 是两条由字符 $\{A,C,G,T\}$ 构成的序列，k 表示字符在序列中出现的位置（这里按习惯从左向右数）。设序列的长度由 L 表示，则 $L_1 = L_2 = 18$。我们看一下，常规统计的频率是怎么得到的。对每一个字符设置一个计数变量，分别为 n_a, n_c, n_g, n_t，然后从左到右扫描字符序列，如果在序列中的位置 k 处，遇到字符 A，则 n_a 加 1，若遇到的是字符 C，则 n_c 加 1，如此下去，直至到达序列结束。然后，用最后所得到的 n_a, n_c, n_g, n_t 分别除以序列的长度 L，即得到各字符在序列中出现的频率。对这两条序列来说，频率都是

$$p_a = \frac{n_a}{L} = \frac{5}{18} \quad p_c = \frac{n_c}{L} = \frac{4}{18} \quad p_g = \frac{n_g}{L} = \frac{5}{18} \quad p_t = \frac{n_t}{L} = \frac{4}{18}。$$

其中，n_a, n_c, n_g, n_t 分别表示字符 A、C、G、T 在序列中出现的次数，即各字符的频度。由于 s_1 和 s_2 的组成是相同的，所以，$p_{i_1} = p_{i_2} \quad (i = A,C,G,T)$。

如果把 s_1 中第 9 个位置处的 T 和第 10 个位置处的 C 互换之后，就得到 s_2，可见 s_1 和 s_2 的区别在于排列方式不同。上面看到，$p_{i_1} = p_{i_2} \quad (i = A,C,G,T)$，即是说，通常的频率无法反映排列信息，因而若只使用它们来表征两个字符串，便无法区分 s_1 和 s_2。那么如何才能反映排列信息呢？科学学中对科学实验的一种分类方法就是把科学实验分成保护性实验和扰动性实验：“保护性实验设法排除对研究对象的干扰，保持其稳定不变，在比较纯粹的状态下反映对象；扰动性实验设法给研究对象施加干扰，造成研究对象的变化，在激发的状态下考察对象”[126]。受此启发，如果我们设法对字符序列施加一个人为的干扰，就有可能达到反映排列信息的目的。

这里，我们采用对位置赋权的方式，即不同的位置被赋予不同的权重，那么我们就得到了字符序列的加权统计，而常规的平权统计则可以看作是它的一个特例。设对字符序列中的

各位置k处赋予的权重用函数$I(k)$表示，再对每一字符设置计数变量分别为I_a, I_c, I_g, I_t，从左到右扫描字符序列，若在位置k处遇到字符A，则I_a加$I(k)$而不是加 1，同样，若遇到C、G、T等字符，I_c、I_g、I_t也分别加$I(k)$而不是加 1，如此下去，直至序列结束，然后将所得的I_a, I_c, I_g, I_t分别除以字符序列的长度L，就得到了各字符的“加权频率”。因为各字符的加权频率与该字符在序列中出现的位置有关，因此，它优于常规的频率之处，就在于它不仅可以反映字符序列组成方面的信息，而且还可以反映排列方面的信息。在本例中，s_1和s_2的区别在于，第 9 个位置处和第 10 个位置处的 C 和 T 互换，只要对$k=9$和$k=10$两个位置赋予不同的权重，即$I(9)\neq I(10)$，则最终所得的I_c和I_t就会不同，即$I_{c_1}\neq I_{c_2}$（和$I_{t_1}\neq I_{t_2}$），从而就可以将s_1和s_2区分开来了。

而上面字符序列的量化方法，把字符序列看成是数的表示，很自然地将位置权重函数引入字符统计，以加权频率来表征字符序列s_1和s_2，就达到了反映排列信息的目的。

2.1.3 在实数域上的推广

仿照自然计数制进行的量化方法，虽然能将长短不一的字符串唯一地对应成一个整数。但正是这种唯一性，造成了量化的“刚性”。前面提到的生物序列分析中的一些基本问题，如：特征提取、序列分类等，都是基于序列之间的相似性的。两个组成上相差很小的字符序列，比如，将上面提到的字符串s_1的第一个字符由 A 变成 T，由于这个变化发生在字符串的左端，造成变化前后的字符串所对应的整数相差很悬殊。这样一来，便无法表现这两个字符串之间的高度相似性了，因此，需要“柔化”一下前面的量化方式。

方法是：把位权函数从以n为底的指数函数，推广到任意实值函数，把$C\to N$映射推广到$C\to X$映射，其中，$X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\,|\,(x_i\in R-\{0\})\}$，即将序数从整数域推广到实数域。

2.2 对偶描述子

对偶描述子是字符序列解析数论模型的核心概念。字符序列所能提供的全部信息不外乎

来自两个方面：组成和排列。字符序列的解析数论模型正是基于推广了的自然计数原理，通过定义“组成权重因子”和“位置权重函数” 共同来反映这两方面的信息。一个“位置权重函数”和一组“组成权重因子”就构成一个**对偶描述子**。

模式描述函数： 对于由字符集 $C=\{c_1,c_2,\cdots,c_i,\cdots,c_n\}$ 中的字符所构成的长度为 L 的字符序列 s，在一对一映射 $f:C\to X$ 映射下，转化成实数序列 x，即

$$\begin{aligned} s&=[s[1],s[2],\cdots,s[k],\cdots,s[L]] \\ &\Downarrow \\ x&=[x[1],x[2],\cdots,x[k],\cdots,x[L]] \end{aligned} \tag{2-4}$$

其中，

$$x[k]=\begin{cases} x_1 & \text{if } (s[k]=c_1) \\ x_2 & \text{if } (s[k]=c_2) \\ & \vdots \\ x_i & \text{if } (s[k]=c_i) \\ & \vdots \\ x_n & \text{if } (s[k]=c_n) \end{cases} \quad (k=1,2,\cdots,L;\ x_i\in X,c_i\in C)$$

对于字符序列 s，定义它的**模式描述函数**如下：

$$N(k)=I(k)\times x[k] \quad (k=1,2,\cdots,L) \tag{2-5}$$

其中，$x[k]$ 前面的系数 $I(k)$，表示位置 k 处被赋予的权重，即为前面提到的位置权重函数。

对偶公式：模式描述函数 $N(k)$ 的前 l 项和公式为

$$S(l)=\sum_{k=1}^{l}N(k)=\sum_{k=1}^{l}I(k)x[k]=\sum_{x_i\in X}x_i\sum_{k_{x_i}}I(k_{x_i}), \tag{2-6}$$

其中，k_{x_i} 表示序列 s 中出现字符 c_i 的位置。余者类推。该公式表示了某种对偶关系，称为“对偶公式”。为了看清对偶公式所表示的对偶关系，下面看两种特殊情形：

1. 若组成平权，即 $x_i=const=1 \quad (x_i\in X)$，则

$$S_{\bar{c}}(l)=\sum_{x_i\in X}1\sum_{k_{x_i}}I(k_{x_i})\approx\int_1^l I(k)dk \tag{2-7}$$

令 $I_{x_i}=\sum_{k_{x_i}}I(k_{x_i})$，则 $S_{\bar{c}}=\sum_{x_i\in X}I_{x_i}$，即 $I_{x_i}\ (x_i\in X)$ 分别表示字符 c_i 对位置权重函数在区间 $[1,\ l]$ 上的积分值的贡献量，称作对偶变量的“排列部”。（实质上是“带位置权重的频度”，

对长度归一化后就是“加权频率”。）

2．若位置平权，即 $I(k)=const=1 \quad (k=1,2,\cdots,L)$，则

$$S_{\bar{p}}(l)=\sum_{x_i\in X} x_i \sum_{k_{x_i}} 1=\sum_{x_i\in X} x_i \sum_{k_{x_i}} 1 \tag{2-8}$$

令 $n_{x_i}=\sum_{k_{x_i}} 1$，则 n_{x_i} 表示字符 c_i 在序列 s 中出现的个数，即 $\sum_{x_i\in X} n_{x_i}=l$。此时有，$S_{\bar{p}}=\sum_{x_i\in X} x_i n_{x_i}$。因为，$n_{x_i}$ 恰为字符 c_i 在字符序列 s 组成中出现的频度，而 x_i 乘在此频度上，表示该频度在整个序列组成中所占的“比重”，故而称 $x_i \quad (x_i\in X)$ 为“组成权重因子”。它们构成了对偶变量的“组成部”。

当然，一般情况下，组成和位置都是不平权的。

奇异模式与标准模式：如果组成和位置都平权，即 $I(k)=\text{constant and } x[k]=\text{constant}$，这时模式描述函数也是常数，即 $N(k)=\text{constant} \quad (k=1,2,\cdots,L)$。不失一般性，设上式中的常数 $\text{constant}=1$，则

$$N(k)=1 \quad (k=1,2,\cdots,L)。 \tag{2-9}$$

组成和位置都平权的模式：$I(k)=\text{constant=1 and } x[k]=\text{constant=1}$，称为**奇异模式**。而把 $N(k)=1 \quad (k=1,2,\cdots,L)$ 的非奇异模式称作**标准模式**。因为，要使 $N(k)$ 等于 1，除了 $I(k)=\text{constant=1 and } x[k]=\text{constant=1}$ 之外，只要 $I(k)=1/x[k] \quad (k=1,2,\cdots,L)$，即 $I(k)$ 和 $x[k]$ 在整个区间上互为倒数就行。标准模式就是指，$N(k)=1$ 且 $I(k)\neq \text{constant and } x[k]\neq \text{constant}, (k=1,2,\cdots,L)$ 的模式。

因为奇异模式下，无法区分各字符，也无法区分各位置，破坏了 s 和 x 之间的等价关系，因此，应该避免奇异模式的出现。标准模式的模式描述函数是一条平行于横轴的直线，其值为 1。一般情况下，标准模式充当参考模式。

对偶变量：称对偶公式的取值 S 为对偶变量，它由两部分组成：

排列部：$I_{x_i} \quad (x_i\in X)$；

组成部：$x_i \quad (x_i\in X)$。

对偶变量与其“组成部”和“排列部”之间关系就是对偶公式：

$$S=\sum_{x_i\in X}x_iI_{x_i} \tag{2-6}$$

不要因名字而造成误解，认为排列部只反映序列的排列信息，而组成部只反映序列的组成信息。其实，“排列部”和“组成部”是密不可分的，没有其中一个就无所谓另一个。它们须联合使用来共同反映组成和排列两方面的信息。

2.3 对偶描述子用于序列特征提取

2.3.1 模式偏离函数与极佳描述

在模式识别中，所谓特征提取是一个从测量空间到特征空间的变换。设测量空间的维数为 n，特征空间的维数为 m，则一般情况下 $m<n$，即该变换为降维变换（支持向量机除外）。若 N 是测量空间，M 是特征空间，则变换 $G:N\to M$ 就是特征提取器。用对偶描述子进行特征提取，是基于样本描述的。引入标准模式作为参考模式，定义如下

模式偏离函数：

$$d=\frac{1}{L}\sum_{k=1}^{L}\big(N(k)-1\big)^2 \tag{2-10}$$

这里1表示标准模式，因此，该函数表示模式 $N(k)$ 与标准模式的偏离程度。用对偶描述子进行特征提取，就是以这个偏离程度作为特征提取器 $G:N\to M$ 优劣的判别准则。偏离程度越小，则变换 $G:N\to M$ 越优。当偏离程度最小，即 d 取最小值时，所给出的模式描述函数 $N(k)$ 称为**最佳描述**。若 $N(k)$ 是 k 的任意函数，则 d 就是定义在一个很宽的函数集上的泛函，而找到最佳描述，就要求解该泛函的极值问题。一般来说，泛函的极值问题要用变分法求解，即拿 d 对 $N(k)$ 作变分，在变分等于 0 处取得极值，但在这里没有必要。因为模式偏离函数是以欧氏距离的方式定义的、以 0 为下确界的二次函数，所以该函数必存在唯一的最小值，且显然在 $N_{\inf}(k)=1$ 处取得。因为，字符序列的模式描述函数 $N(k)$ 被定义成 $N(k)=I(k)\times x[k]\quad(k=1,2,\cdots,L)$，因此，只要 $I(k)$ 和 $x[k]$ 在整个区间上都互为倒数关系，即满足 $I(k)=1/x[k]\quad(k=1,2,\cdots,L)$，就能保证取得 d 的最小值 $d_{\inf}=0$。看下例：

s	A	G	C	A	T	A	G	G	T	C	C	A	C	A	G	T	T	G
$x[k]$	1	3	2	1	4	1	3	3	4	3	3	1	3	1	3	4	4	3
$I(k)$	1	$\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{1}{4}$	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$	1	$\frac{1}{3}$	1	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{3}$
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

。

要描述的字符序列 s 是固定的。在给定一个映射 $f:C\to X$，即给定一组组成权重因子 $x_i\quad(x_i\in X_0)$ 的情况下（上例中 $x_a=1,x_c=2,x_g=3,x_t=4,X_0=\{1,2,3,4\}$），$s$ 转化成实数列 $x[k]$。在一对一映射 $f:C\to X$ 下，实数列 $x[k]$ 和原字符序列 s 之间存在一一对应关系。为保证 d 取最小值 0，位置权重函数 $I(k)$ 在各个 k 点的取值应为相应 $x[k]$ 的倒数，从而 $I(k)$ 也就完全确定了，我们把此时的 $I(k)$ 写作 $I_{\inf}(k)$，即 $I_{\inf}(k)=1/x[k]\quad(k=1,2,\cdots,L)$。$I_{\inf}(k)$ 是以 $x[k]$ 为模板，利用倒数关系“反刻”出来的位置权重函数，它和 $x[k]$ 存在一一对应关系。根据等价关系递推性，$I_{\inf}(k)$ 便构成原序列的等价表示，即无误差的表示。特征提取问题，作为一个从高维到低维的变换，在这里表述为寻找 $I_{\inf}(k)$ 的一个逼近 $I(k)$。

设 $I(k)$ 可以用基 $b(k)$ 展开如下

$$I(k)=\sum_{\gamma}a_\gamma b_\gamma(k)\quad(\gamma=1,2,\cdots,m) \tag{2-11}$$

式中，a_γ 与 k 无关，方程右边是 m 项的和，$m\le L$，L 是序列长度。用逼近函数 $I(k)$ 代替 $I_{\inf}(k)$，就可能引入误差。由于 $I_{\inf}(k)$ 能令模式偏离函数取得其下确界 $d_{\inf}=0$，若设 $I(k)$ 使模式偏离函数的取值为 d，则用 $I(k)$ 逼近 $I_{\inf}(k)$ 所引起的误差恰为 $d-d_{\inf}=d-0=d$，这就是能以 d 作为特征提取器 $G:N\to M$ 的优劣的判别准则的原因。关于基 $b(k)$ 的选取，后面还有详细的讨论。在给定一组组成权重因子 $x_i\quad(x_i\in X_0)$ 的情况下，选定基 $b(k)$ 后，就可以应用 d 取极小值的条件确定相应的系数 a。具体做法为：

由 $\dfrac{\partial d}{\partial a_\gamma}=0$ 得，
$$\begin{aligned}u_{\alpha\beta}&=\sum_{k=1}^{L}b_\alpha(k)b_\beta(k)x[k]^2\\ v_\alpha&=\sum_{k=1}^{L}b_\alpha(k)x[k]\end{aligned}\qquad(\alpha,\beta=1,2,\cdots,m) \tag{2-12}$$

其中，$b_\alpha(k)$ 和 $b_\beta(k)$ 分别为第 α 和第 β 个基，$x[k]\in X_0$ 是实数列 x 的第 k 个位置处的元素。逼近函数 $I(k)$ 展开式的系数向量 **a** 可由矩阵 **u** 和向量 **v** 求出：

$$\mathbf{a}=\mathbf{u}^{-1}\mathbf{v} \tag{2-13}$$

其中，$\mathbf{a}=(a_1,a_2,\cdots,a_\gamma,\cdots,a_m)$，而矩阵 **u** 和向量 **v** 的各元素由 $u_{\alpha\beta}$ 和 v_α 给出。

上面给的模式偏离函数的定义是针对单个字符序列的。若要提取多条字符序列的公共特征，则把多条字符序列总体的模式偏离程度定义成各单个字符序列的模式偏离函数之和：

$$D=\frac{1}{n}\sum_{j=1}^{n}d_j \tag{2-14}$$

其中，n 是序列的数目；$d_j=\frac{1}{L_j}\sum_{k=1}^{L_j}(N_j(k)-1)^2$ 是第 j 个序列的模式偏离函数，L_j 是第 j 条序列的长度，$N_j(k)$ 是第 j 条序列的模式偏离函数。

在给定映射 $f:C\to X$，即给定一组组成权重因子 x_i $(x_i\in X_0)$ 的情况下，

由 $\frac{\partial D}{\partial a_\gamma}=\sum_{j=1}^{n}\frac{\partial d_j}{\partial a_\gamma}=0$ 得，$\begin{matrix}U_{\alpha\beta}=\sum_{j=1}^{n}u_{\alpha\beta}^{j}\\ V_\alpha=\sum_{j=1}^{n}v_\alpha^{j}\end{matrix}$ $(\alpha,\beta=1,2,\cdots,m)$。 (2-15)

此时，展开式(2-11)的系数向量 **a** 可由矩阵 **U** 和向量 **V** 表示出来：

$$\mathbf{a}=\mathbf{U}^{-1}\mathbf{V} \tag{2-16}$$

而矩阵 **U** 和向量 **V** 分别就是 n 个单序列情形下的矩阵 **u** 和向量 **v** 的和。

利用极值条件确定的系数向量 **a** 就是代表**原字符序列的特征量**，它相应的逼近函数写作 $I^*(k)$，模式描述函数写作 $N^*(k)$，分别称为**极佳位置权重函数**和**极佳描述**。极佳描述就是对应于模式偏离函数极小值 d^* 的模式描述函数，而相应的组成权重因子和位置权重函数的整体就称为**极佳对偶描述子**。

2.3.2 序列的重构与失真度量

利用（2-13）式求出展开式系数向量 **a** 之后，逼近函数 $I(k)$ 就完全确定了。此时，位

置权重函数可以根据（2-11）式写出来。那么逼近的效果如何度量呢？除了应用前面的模式偏离函数值之外，还可以从序列重构的失真程度来度量。下面看如何重构原字符序列。

根据（2-11）式的位置权重函数 $I(k)\quad (k=1,2,\cdots,L)$，可以通过取其倒数的方式，构造出一个实数列 $x'[k]\quad (k=1,2,\cdots,L)$。$x'$ 中的各元素，可能并不属于原来的组成权重因子的集合 X。为了能够应用 $f:C\to X$ 的逆映射 $f^{-1}:X\to C$ 重构出原字符序列，需要对实数列 x' 进行一下处理，使 x' 中的各元素属于 X。这个处理过程可根据下式进行：

$$\begin{aligned} &x[k]=x_i \\ &i=\arg\min_{x_i\in X}(\left|x'[k]-x_i\right|) \end{aligned}\quad (k=1,2,\cdots,L;\ x_i\in X) \tag{2-17}$$

其中，$x[k],\ (k=1,2,\cdots,L)$ 是处理过之后的实数序列，它是由 X 中的元素，即原组成权重因子组成的。而对于位置 k 处，究竟取哪个组成权重因子，即如何确定 i，则是由 $x'[k]$ 与各组成权重因子之间的距离决定的，$x'[k]$ 与哪个 x_i 最接近，就取哪个 x_i 作为处理过之后的序列 $x[k]$ 中的元素。有了实数列 $x[k]\quad (k=1,2,\cdots,L)$，就可以根据逆映射 $f^{-1}:X\to C$ 得到字符序列 s'。s' 是根据提取的特征向量 **a** 重构出来的，它与原字符序列 s 之间的 Hamming 距离为：

$$d_{\mathrm{hm}}=\sum_{k=1}^{L}\delta(s'[k]-s[k])\ ,\ \delta(s'[k]-s[k])=\begin{cases}1 & \text{if } s'[k]=s[k]\\ 0 & \text{if } s'[k]\neq s[k]\end{cases} \tag{2-18}$$

它表示重构出来的字符序列 s' 与原字符序列之间的不相同的字符的个数。据此，定义重构的失真率为：

$$e=\frac{d_{\mathrm{hm}}}{L} \tag{2-19}$$

即不相同的字符的个数在字符序列的长度中所占的比率。

2.3.3 基函数的选择

在公式(2-11)中，假设位置权重函数可以用基 $b(k)$ 展开，即

$$I(k)=\sum_{\gamma}a_\gamma b_\gamma(k)\quad (\gamma=1,2,\cdots) \tag{2-20}$$

那么，究竟怎样选取逼近基函数 $b(k)$ 呢？方法是多种多样的。不同的选取方法对应于不同的变换。

（1）**周期基**。**最佳一致逼近理论**的基本定理之一，外尔斯特拉斯第二定理表述如下：

定理 2.1 设 $f(x)\in C_{2\pi}$，那么对于任意给定的 $\varepsilon>0$，都存在这样的三角多项式 $T(x)$，使得

$$\max_{-\pi\le x\le\pi}\left|T(x)-f(x)\right|<\varepsilon \tag{2-21}$$

其中，$C_{2\pi}$ 表示 $(-\infty,+\infty)$ 一切具有 2π 周期的连续实函数的集合；$T(x)$ 为三角多项式

$$T(x)=a_0+\sum_{k=1}^{n}(a_k\cos kx+b_k\sin kx) \tag{2-22}$$

由定理 2.1 知，三角多项式可以以任意精度逼近任意 $C_{2\pi}$ 类函数。可见，三角多项式有很强的逼近能力[127]。但是，对于字符序列分析的特定问题而言，位权函数为定义在正整数区间上的离散函数（自变量为 k），其周期只可能是整数周期。所以，为了使周期基有意义，三角多项式，选择如下形式，

$$I(k)=a_0+\sum_{m=1}^{n}(a_m\cos\frac{2\pi k}{m}+b_m\sin\frac{2\pi k}{m}) \tag{2-23}$$

由于，正弦函数在整周期处很容易取得零值。所以在实际应用时，只用余弦函数就够了，通常取成

$$I(k)=\sum_{m=1\ (\text{or }2)}^{n}a_m\cos\frac{2\pi k}{m}\quad(n\le L) \tag{2-24}$$

式中 L 为待处理的序列的长度；m 对应于 m 周期性。这里谈一下 $m=1$ 的项。$m=1$ 对应于 1 周期性，因为 k 只能取正整数值，而对整数而言，1 周期性很特殊。在整数区间上，1 周期性其实就是无周期性，相当于在区间上处处可以取同一个值，这对于对偶描述子的特征提取是无碍的，相当于在位置权重函数的展开式中引入了一个阈值。但是，却对后面将要提到的对偶描述子的**交替式学习**不利，因为它可能导致奇异模式的出现。因此，在对偶描述子的交替式学习过程中，令 m 从 2 开始，而不是从 1 开始。（详见“对偶描述子的学习”节）。

对于整数而言，构造周期函数还有一个更简单的方法，就是采用取模运算。$f(k)=k\bmod m$ 即是一个以 m 为周期的函数。为了使位置权重的取值不为零，通常可以取作如下形式 $I(k)=e^{k\bmod m}$。另外，为了减少冗余，也可以更进一步地，令 m 只取素数值。

（2）**小波基**。 对于具有紧支集并满足零均值条件

$$\int_{-\infty}^{+\infty}\psi(t)dt=0 \tag{2-25}$$

的母小波函数，进行伸缩和平移就得到具有局域特征和多尺度分析能力的小波基函数

$$\psi_{a,b}(t)=|a|^{-\frac{1}{2}}\psi(\frac{t-b}{a}) \tag{2-26}$$

式中$b\in R,a\in R-\{0\}$，$\psi_{a,b}(t)$中的参变量a反映函数的尺度（或宽度），变量b检测沿t轴的平移位置[128]。

位权函数可用小波基展开如下

$$I(k)=\sum_{\alpha}\sum_{\beta}a_{\alpha,\beta}\psi(\alpha k-\beta) \tag{2-27}$$

当取$\alpha=2^j$，$\beta=n$时，就是通常的离散二进小波，而此时，$a_{\alpha,\beta}=\langle I(k),\psi_{\alpha,\beta}(k)\rangle$。

而在实际使用中，可以取$\alpha=m$，$\beta=\frac{n}{m}L$，$m,n\in Z^+$，L为序列的长度。即

$$I(k)=\sum_{m}\sum_{n}a_{m,n}\psi(mk-\frac{n}{m}L)\quad(m\in[1,L],n\in[0,m]) \tag{2-28}$$

其实，就是将原字符序列m等分（采样m个点），在每个采样点处做一个小波基，并在这些小波基上展开$I(k)$。其中，m决定了采样个数，同时也决定了小波的支集长度，起到了尺度因子的作用；n则表示沿序列的平移，n从0取到m，恰好盖满原序列。

另一种方法，就是固定采样的间隔m，每次从左向右移动m个位置，直至盖满全序列，即

$$I(k)=\sum_{m}\sum_{n}a_{m,n}\psi(mk-nm)\quad(m\in[1,L],n\in[0,\frac{L}{m}]) \tag{2-29}$$

这里没有使用小波分析中标准的分解与重构算法，即 Mallat 算法（也叫快速小波变换算法 FWT），因为，该算法与后面将给出的“立体描述”在很大的程度上是重合的。不过，这里通过选取不同的m值，也能达到多尺度逼近的效果。

与小波分析方法不同的是，这里展开式系数$a_{m,n}$的确定，可以通过解线性方程组得到，此线性方程组就是根据模式偏离函数取极小值的条件——偏微分等于0，而得到的方程组。而在小波分析理论中，系数$a_{m,n}$是通过计算原始信号与小波滤波器的卷积而得到的[129]。

（3）**杂合基**。杂合基取作周期基和小波基的乘积。其中，周期基反映全局的周期性振荡而小波基反映局部的信号瞬变。即

$$I(k)=\sum_{m}\sum_{n}a_{m,n}\psi(mk-\frac{n}{m}L)\cos\frac{2\pi k}{m}\quad(m\in[1,L],n\in[0,m]) \tag{2-30}$$

或者

$$I(k)=\sum_{m}\sum_{n}a_{m,n}\psi(mk-nm)\cos\frac{2\pi k}{m}\quad(m\in[1,L],n\in[0,\frac{L}{m}]) \tag{2-31}$$

式中$\cos\frac{2\pi k}{m}$可代之以$e^{k \bmod m}$。这里的杂合基大体相当于小波包或者局部余弦基(Malvar 小波)[130]。

2.4 对偶描述子的交替式学习

极佳组成权重因子 在“对偶描述子用于模式特征提取”一节中，在给定映射$f:C\to X$，即给定一组组成权重因子$x_i\quad(x_i\in X_0)$后，用计算模式偏离函数极小值的办法去逼近位置权重函数，从而得到了用逼近函数的系数向量来表示的字符序列的特征。然而，那只是问题的一个方面。另一方面，如果预先给定一个位置权重函数$I_0(k)\quad(k=1,2,\cdots,L)$，就是说，对要描述的字符序列$s$中的每个位置$k$赋权，权重用函数$I_0(k)$表示，则相当于对原字符序列施加了一个人为的干扰。在此干扰下，观察字符序列所做出的反应。这时，如果还要求模式偏离函数取得其最小值$d_{\inf}=0$，则可以根据$I_0(k)\quad(k=1,2,\cdots,L)$的取值，“反刻”出一个实数列$x[k]$来，然而，如果$I_0(k)$是任意给的，那么，反刻出来的$x[k]$未必是有效的。看下面这个例子：

s	A	G	C	A	T	A	G	G	T	C	C	A	C	A	G	T	T	G
$x[k]$	1	3	2	1	4	1	3	2	4	9	2	1	2	1	3	4	4	3
$I_0(k)$	1	$\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{1}{4}$	1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{9}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{3}$
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

要描述的字符系列是固定的。预先给定的位权函数$I_0(k)$如上例所示，看由$d_{\inf}=0$“反刻”出来的$x[k]$。在$k=2$和$k=8$处，字符序列s中的出现的字符都是G，可是由$I_0(k)$“反刻”出来的$x[k]$，在此两处分别等于3和2，那么，让G映射到3呢还是映射到2呢？这就出现了矛盾。同样，$k=3$和$k=10$处的字符C也有矛盾。这样一来，为了模式偏离函数取得极小值，就得在$G\to 3$和$G\to 2$之间作一个折衷，同样，对于字符C，也要在$C\to 2$和$C\to 9$之间做一个折衷。与此同时，又要满足映射$f:C\to X$是一对一映射的条件，即字

符 G 和字符 C 不能对应同一个数。由于以上原因，在任给一个 $I_0(k)$ 时，模式偏离函数能取得极小值，但达不到其下确界。使模式偏离函数极小的组成权重因子 x_i $(x_i \in X^*)$，称为**极佳组成权重因子**，可如下求出：

$$\text{由}\frac{\partial d}{\partial x_i}=0 \text{ 可得，} x_i=\frac{\sum\limits_{k_{x_i}} I_0(k_{x_i})}{\sum\limits_{k_{x_i}} I_0^2(k_{x_i})} \quad (x_i \in X^*)\text{，} \tag{2-32}$$

式中， k_{x_i} 表示实数列 x 中出现组成权重因子 x_i 的那些位置。

同样，在多序列情形下， $D=\frac{1}{n}\sum\limits_{j=1}^{n} d_j$ ，若 $I(k)$ 已知，则由 $\frac{\partial D}{\partial x_i}=\sum\limits_{j=1}^{n}\frac{\partial d_j}{\partial x_i}=0$ 可得，

$$x_i=\frac{\sum\limits_{j}\sum\limits_{k_{x_i}} I(k_{x_i})}{\sum\limits_{j}\sum\limits_{k_{x_i}} I(k_{x_i})} \quad (x_i \in X^*)\text{。} \tag{2-33}$$

这组响应人为干扰 $I_0(k)$ 的极佳组成权重因子，是根据 d 值极小“挤压”出来的，也可以作为原字符序列的特征，它和前面提到的极佳位置权重函数一起，构成极佳对偶描述子。

综合以上两方面，我们看一下“对偶”的含义。因为最佳描述 $N_{\inf}(k)=1$， $I(k)$ 与 $x[k]$ 以倒数关系互补，因此，依据要描述的字符序列，若给定映射 $f:C\to X$ ，即给定了一组组成权重因子后，就可以以实数列 $x[k]$ 为模板“反刻”出一个位置权重函数 $I(k)$，反之，若给定一个人为的干扰，即给定了一个位置权重函数 $I(k)$ 后，则可以由 d 值极小“挤压”出一组组成权重因子 $x_i(x_i \in X)$，因此， $I(k)$ 和 x_i 之间，此消彼长，互为凹凸，这也就是“对偶”的含义。

交替式学习过程 对偶描述子的学习，就是将组成权重因子看作一组参数，并把逼近函数 $I(k)$ 在一组基上展开成一个含有参数的表达式，之后利用多元函数取极值的条件，即偏微分等于 0，确定各参数的值。现在把前头提到的两种情形，一并开列如下：

情形 1：若映射 $f:C\to X$ 已知，即预先给定了一组组成权重因子 x_i $(x_i \in X_0)$，则

由 $\frac{\partial d}{\partial a_\gamma}=0$ 得，$\begin{aligned} u_{\alpha\beta} &= \sum_{k=1}^{L} b_\alpha(k) b_\beta(k) x[k]^2 \\ v_\alpha &= \sum_{k=1}^{L} b_\alpha(k) x[k] \end{aligned} \quad (\alpha,\beta=1,2,\cdots,m)$ (2-12)

逼近函数 $I^*(k)$ 展开式的系数向量 $\mathbf{a}$ 可由矩阵 $\mathbf{u}$ 和向量 $\mathbf{v}$ 求出：

$$\mathbf{a}=\mathbf{u}^{-1}\mathbf{v} \tag{2-13}$$

情形 2：若预先给定位权函数 $I_0(k)\quad (k=1,2,\cdots,L)$，即是说，通过位置赋权的方式，对字符序列施加一个人为的干扰，则

由 $\frac{\partial d}{\partial x_i}=0$ 可得，$x_i=\frac{\sum_{k_{x_i}} I_0(k_{x_i})}{\sum_{k_{x_i}} I_0^2(k_{x_i})} \quad (x_i\in X^*)$ (2-32)

式中， k_{x_i} 表示实数列 x 中出现组成权重因子 x_i 的那些位置。

上面列出的两种情形的任何一种，都形成对偶描述子的“一次性学习方式”，即根据已知的组成权重因子或位置权重函数这两者之一，利用对偶关系一次性地求出另一个。此外，对偶描述子还存在另一种学习方式——“交替式学习”。

考虑下面的问题：如果在情形 1 下，由已知的组成权重因子求出了逼近函数 $I(k)$ 展开式中的各系数，即确定了位置权重函数的一个估计，接下来，再以这个求得的逼近函数 $I(k)$ 为前提进入情形 2，并由此求得一组组成权重因子 x_i，那么，这组 x_i 与情形 1 中预设的那组 $x_i\quad (x_i\in X_0)$ 相同吗？如果不相同，则学习过程还可以进行下去，即再以这组新获得的组成权重因子 x_i 为前提进入情形 1，继续求位权函数的逼近 $I(k)$，……，如此交替，结果如何呢？

数字实验表明，在交替式学习过程中，模式偏离函数的取值 d 还会进一步下降，以 d 值为标志的极佳描述 $N^*(k)$ 是唯一的，即最终取得的 d^* 值是唯一的，它只决定于字符序列本身的特点和基 $b(k)$ 的选取方式，而与组成权重因子和位置权重函数展开式系数的初始值都没有关系，也和进入交替式学习过程的次序（即先由情形 1 进入还是先由情形 2 进入）没有关系。虽然最终取得的 d^* 是唯一的，但是，极佳对偶描述子却不是唯一的，即最终获得的

组成权重因子 x_i 和位置权重函数 $I(k)$（其展开式的系数）不是唯一的，它们依赖于其初始值和进入交替式学习过程的次序。这不难理解，由于 $N(k)$ 被定义成 $I(k)$ 与 $x[k]$ 的乘积，由一个积不能确定它的两个因子，因此，与 $I(k)$ 和 $x[k]$ 对应的极佳对偶描述子不是唯一的。

交替式学习过程的特点 对偶描述子的交替式学习过程就是不断调整组成权重因子和位置权重函数，以逐步逼近模式偏离函数极小值的过程。最终，模式偏离函数收敛于全局极小，而组成权重因子和位置权重函数则收敛于某一个极佳描述子。交替式学习过程的进行是有条件的，就是说，在交替式学习过程中，各字符的组成权重因子不能够趋于同一个常数，同时，位置权重函数也不能趋于一个常数，即在逼近标准模式的过程中要避开奇异模式，这也正是前面提到的为什么三角多项式要从 2 开始而非 1 开始的原因。

对偶描述子的学习过程是一个无约束优化过程，具有以下两个特点：（1）**很强的一致收敛性**。因为模式偏离函数是二次型的函数，所以，它存在唯一的全局极小值。不论对偶描述子的初始参数是什么，模式偏离函数最终总能收敛到这个唯一的全局极小值。因为没有局部极小的干扰，所以对偶描述子的学习过程是一致收敛的。（2）**收敛速度快**。上面说过，模式偏离函数的极小值是唯一的，它只依赖于序列本身的特点和基函数的选取，而与对偶描述子参数的初始值没有关系。然而，对应于该极小值的极佳对偶描述子却不是唯一的，它依赖于参数的初始值，这是模式描述函数被设计成两项乘积的结果。正是极佳对偶描述子的这种不唯一性，赋予了它随遇平衡的能力。因为不存在唯一的极佳对偶描述子，对偶描述子在学习过程中，就没有必要长途跋涉去奔向一组唯一的参数，而只要在其初始值附近，因利就便，找到一个可使模式偏离函数达到全局极小的极佳对偶描述子就行了。故此，收敛速度较快。

2.5 对偶描述子用于序列识别

上面的对偶描述子学习的结果是得到一个模式偏离函数极小值和一个极佳对偶描述子。那么它们有什么用呢？事实上，对偶描述子学习的过程，就是逼近样本的最佳描述的过程。学习的结果就是基于样本的描述而提取的序列的特征。打个比方，学习过程就是对要描述的序列量体裁衣，而获得的极佳对偶描述子就是做成的一件衣服。这件衣服是按照要描述的序列做的，其尺码、肥瘦恰好反映了原序列的特征。序列识别就是让待识别的序列试穿这件衣服，如果合适，说明待识别序列与原序列之间特征相近，如果不合适，则说明两条序列之间相去较远。通常情况下，用对偶描述子提取的是一类（多条）字符序列的共同特征，这样所

得到的极佳对偶描述子就是反映该类字符序列共同特征的一件衣服。如果待识别序列穿着合适，就可以把它归到这一类中来。

下面按“一次性学习”和“交互式学习”两种情况，分别讨论如何运用对偶描述子进行序列的识别。

1. 一次性学习。通常是给定一组组成权重因子，然后根据模式偏离函数极小，求得位置权重函数的一个逼近 $I^*(k)$，则该位置权重函数就携带了序列的特征信息。对于待识别的字符序列 s'，用函数 $I^*(k)$ 对各位置加权，即把 $I^*(k)$ 作为一个人为干扰引入统计过程。假定组成平权，按照对偶公式提取对偶变量的排列部，归一化后，作为序列的特征量。具体做法为：从左到右扫描待识别序列 s'，设在 s' 中的位置 k 遇到字符 c_i，则令与字符 c_i 对应的计数变量增加 $I^*(k)$，直至序列结束，然后将所得的对偶变量的排列部 I_{x_i} 除以序列 s' 的长度后，作为原序列的特征量。因为，对偶变量的排列部表示加权的频度，它归一化后就表示加权频率，能够反映组成和排列两方面的信息，所以可以作为原字符序列的识别变量。有了识别变量，就可以结合其它的一些判别方法，如 Fisher 判别，基于正负两类样本来训练参数，进行序列的识别了。

2. 交替式学习。交替式学习的结果对应于模式偏离函数的全局极小。它所生成的极佳对偶描述子充分携带了原序列的特征信息。而对于待识别序列，只要以此极佳对偶描述子从头到尾描述一遍，计算它的模式偏离函数值，如果所得的 d 值足够小，小于某个预设的阈值，就可以把它和原序列归为同类。具体做法为：从左到右依次检查待识别序列 s'，设在 s' 中的位置 k 处遇到字符 c_i，则用与字符 c_i 对应的组成权重因子 x_i 乘以位置权重函数 I^* 在此处的取值 $I^*(k)$，将所得的积减去 1 后取平方，并累加到模式偏离函数的取值 d 上，如此直至序列结束，然后，拿最后的 d 值和预设的阈值作比较，若小于阈值，则把待识别序列与原序列归为同类，否则，不同类。

以上便是用对偶描述子进行序列识别的具体做法。至于阈值如何选取，应该在高识别率和低伪正率之间做个折衷，关于识别率和伪正率等概念，详见第四章和第五章。

上面提到的都是一类（两类）识别问题。多类识别如何处理呢？因为对偶描述子是基于单类的样本描述来提取特征的，因此，对于多类问题，要对其中的每个单类分别训练对偶描

述子。还按前面的两种学习方式简单说一下。在一次性学习中，仅是应用对偶描述子的排列部作为序列的特征量，因此，不区分单类多类，它的单类和多类问题的区别，是由与它一起使用的辅助识别算法来解决的。比如，可以用对偶描述子的归一化后的排列部（加权频率）作为特征量，对一批字符序列进行聚类分析等。而在交替式学习的情形下，则分别对各类训练对偶描述子，然后，分别用各类的极佳对偶描述子去描述待识别序列，求得多个 d 值，哪个 d 相对最小，就把待识别序列归到哪一类。设用第 i 类的对偶描述子描述待识别序列所得的模式偏离函数值为 d_i，则待识别序列所属的类别 i 由下式给出：

$$i = \arc\min(d_i) \tag{2-34}$$

2.6 矢量形式、几何表示与应用扩展

2.6.1 矢量形式

在第一章“粗粒化与字符表示”部分，提到用字符表示一个物理实体是假定各物理实体间彼此为全同粒子，并且不再去追究它的内在属性，而是以符号代之。然而，实际上，物理实体是具有多种属性的。比方说，每种氨基酸都是一个物理实体，具有多方面的属性，诸如，惯性、电性、疏水性等等；把每一个属性量化，都对应一个物理量。前面介绍的对偶描述子，每一个字符对应一个实数，是其标量形式，因此，只能刻画一个物理量（或它的一个分量）。如果要同时刻画多个物理量，这种标量形式就不敷使用了。因此，有必要提出对偶描述子的矢量形式。对偶描述子的矢量形式是其标量形式的直接推广。在标量形式中，每一个字符对应一个数，而在矢量形式中，每一个字符对应一个矢量。矢量的每一个分量，对应于一个物理属性，而分量的取值，就是这一属性的量值。此外，矢量化后的对偶描述子，包含了更多的参数，在有必要的时候，可以去调整这些参数以逼近一个映射关系。

进一步将 $C \to X$ 映射推广到 $C \to \mathbf{X}$ 映射，其中，$C = \{c_1, c_2, \cdots, c_i, \cdots, c_n\}$ 为 n 个字符组成的集合，而 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_n \mid \mathbf{x}_i \in R^m\}$ 为 m 维实矢量的集合。矢量集 $\mathbf{X}$ 中的矢量为列矢量，称作“编码矢量”。把矢量集 $\mathbf{X}$ 中的这 n 个 m 维的列矢量排放在一起，就形成一个 m 行 n 列的矩阵 $M_{m\times n}$，称作“编码矩阵”，即

$$M_{m\times n}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1i} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2i} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{j1} & x_{j2} & \cdots & x_{ji} & \cdots & x_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mi} & \cdots & x_{mn} \end{bmatrix} \quad (x_{ji}\in R-\{0\}; i=1,2,\cdots,n, j=1,2,\cdots,m)$$

该矩阵中的每一列，对应字符集C中的一个字符，每一行则是矢量的一个分量。为了减少信息重复，有如下原则需要注意：

原则 1 应使各分量彼此独立，即应使“编码矩阵”的行矢量之间彼此线性无关，最好是能彼此正交。（这一原则并非必然要求。）

这一原则是说，编码矢量的各分量应该尽量反映不同方面的属性，从而总体上给出一个更完善的描述。若编码矩阵的行向量之间彼此线性无关，便不可能由其中一些作线性组合生成另一些，从而使各分量在描述字符序列时，能够尽量减少信息的重复。进一步，如果行向量之间能够彼此正交，则信息的冗余度降到最低，定义此时，对偶变量的序列所给出的字符序列的描述是**无冗余表示**。

在$C\to\mathbf{X}$的$n!$种“一对一”的映射方式中，仍然称如下这种映射方式

$$f_1: C \to \mathbf{X} = f_1: c_i \to \mathbf{x}_i = \begin{matrix} \{\ c_1\ \ c_2\ \cdots\ c_i\ \cdots\ c_n\ \} \\ \updownarrow\ \updownarrow\ \cdots\ \updownarrow\ \cdots\ \updownarrow \\ \{\ \mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_i\ \cdots\ \mathbf{x}_n\ \} \end{matrix}$$

为映射f_1。

在映射f_1下，长度为L的字符序列

$$s=[s[1],s[2],\cdots,s[k],\cdots,s[L]]\ (s[k]\in C,\ k=1,2,\cdots,L)$$

转化成m维实矢量的序列

$$\mathbf{x}=[\mathbf{x}[1],\mathbf{x}[2],\cdots,\mathbf{x}[k],\cdots,\mathbf{x}[L]],\quad (\mathbf{x}[k]\in\mathbf{X})$$

其前l项带权和公式，即对偶公式如下

$$\mathbf{s}(l)=\sum_{k=1}^{l}\mathbf{I}(k)\mathbf{x}[k]\quad (l=1,2,\cdots,L) \tag{2-35}$$

式中

$$\mathbf{I}(k)=\begin{bmatrix} I_{11}(k) & I_{12}(k) & \cdots & I_{1\beta}(k) & \cdots & I_{1m}(k) \\ I_{21}(k) & I_{22}(k) & \cdots & I_{2\beta}(k) & \cdots & I_{2m}(k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_{\alpha 1}(k) & I_{\alpha 2}(k) & \cdots & I_{\alpha\beta}(k) & \cdots & I_{\alpha m}(k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_{m1}(k) & I_{m2}(k) & \cdots & I_{m\beta}(k) & \cdots & I_{mm}(k) \end{bmatrix}_{m\times m} \tag{2-36}$$

为**位置权重函数矩阵**（m 阶方阵），其中每个元素 $I_{\alpha\beta}(k)$ 都是一个位置权重函数，$\mathbf{x}[k]$ 为矢量序列 $\mathbf{x}$ 的第 k 个矢量（维数为 m），对应于字符序列 s 的第 k 个字符：

$$\mathbf{x}[k]=\begin{cases} \mathbf{x}_1 & \text{if } (s[k]=c_1) \\ \mathbf{x}_2 & \text{if } (s[k]=c_2) \\ & \vdots \\ \mathbf{x}_i & \text{if } (s[k]=c_i) \\ & \vdots \\ \mathbf{x}_n & \text{if } (s[k]=c_n) \end{cases} \quad (k=1,2,\cdots,L;\ \mathbf{x}_i\in\mathbf{X},c_i\in C)$$

由公式（2-35）可知，当 l 从1变到 L 时，可得到对偶变量 $\mathbf{s}(l)$ 的一个序列 $\mathbf{s}$，它也是一个矢量序列，即 $\mathbf{s}=[\mathbf{s}[1],\mathbf{s}[2],\cdots,\mathbf{s}[l],\cdots,\mathbf{s}[L]]$，该序列就是编码矢量序列 $\mathbf{x}$ 的前 l 项带权和序列。

模式偏离函数：

$$d=\frac{1}{L}\sum_{k=1}^{L}\left|\mathbf{N}(k)-\mathbf{c}\right|^2=\frac{1}{L}\sum_{k=1}^{L}\left|\mathbf{I}(k)\mathbf{x}[k]-\mathbf{c}\right|^2 \tag{2-37}$$

式中 $\mathbf{I}_{m\times m}(k)$ 为前面的位置权重函数矩阵（2-36），其中每个元素 $I_{\alpha\beta}(k)$ 都是一个位置权重函数，并可用基 $b(k)$ 展开成 l 项的和如下

$$I_{\alpha\beta}(k)=\sum_{\gamma=1}^{l}a_{\gamma}^{\alpha\beta}b_{\gamma}^{\alpha\beta}(k) \tag{2-38}$$

而 $\mathbf{x}[k]$ 为矢量序列 $\mathbf{x}$ 的第 k 个矢量（维数为 m），对应于字符序列 s 的第 k 个字符；$\mathbf{c}$ 为 m 维常矢量。

交替式学习：

若位权函数矩阵 $\mathbf{I}_{m\times m}(k)$ 已知，

由 $\frac{\partial d}{\partial \mathbf{x}_i}=0$ 可得，$\mathbf{x}_i=\mathbf{w}^{-1}\mathbf{y}$， (2-39)

其中，矩阵 $\mathbf{w}$ 和矢量 $\mathbf{y}$ 的各元素为

$$\begin{aligned} w_{pq}&=\sum_{k_{\mathbf{x}_i}}\sum_{\alpha=1}^{m} I_{\alpha p}(k_{\mathbf{x}_i})I_{\alpha q}(k_{\mathbf{x}_i}) \\ y_p&=\sum_{k_{\mathbf{x}_i}}\sum_{\alpha=1}^{m} c_\alpha I_{\alpha p}(k_{\mathbf{x}_i}) \end{aligned} \qquad (\alpha,p,q=1,2,\cdots,m)。 \tag{2-40}$$

若 $f:C\to\mathbf{X}$ 已知，

由 $\frac{\partial d}{\partial \mathbf{a}_\alpha}=0$ 得，$\mathbf{a}_\alpha=\mathbf{u}_\alpha^{-1}\mathbf{v}_\alpha$， (2-41)

其中，$\mathbf{a}_\alpha$ 表示位置权重函数矩阵的第 α 行的所有元素展开式的总的系数行向量，该行向量包含 $m\times l$ 个分量，这些分量每 l 个一组，分别属于第 α 行的 m 个元素，而矩阵 $\mathbf{u}_\alpha$ 和向量 $\mathbf{v}_\alpha$ 的元素则由下式给出：

$$\begin{aligned} u_{pq}^{\alpha}&=\sum_{k=1}^{L} b_{\beta\gamma}^{\alpha}(k)b_{\beta'\gamma'}^{\alpha}(k)x[k]_s x[k]_t \\ v_p^{\alpha}&=c_\alpha\sum_{k=1}^{L} b_{\beta\gamma}^{\alpha}x[k]_s \end{aligned} \qquad (\alpha,p,q=1,2,\cdots,m) \tag{2-42}$$

$$\beta=\left\lceil\frac{p}{l}\right\rceil \quad \gamma=\delta(p \bmod l)\cdot l+p \bmod l;$$

其中，$\beta'=\left\lceil\frac{q}{l}\right\rceil \quad \gamma'=\delta(q \bmod l)\cdot l+q \bmod l;$ (2-43)

$$s=\left\lceil\frac{p}{l}\right\rceil \quad t=\left\lceil\frac{q}{l}\right\rceil$$

式中 $\lceil\square\rceil$ 表示取不小于“$\square$”的最小整数，$\delta(\square)=\begin{cases}1 & \text{if } \square=0\\ 0 & \text{if } \square\neq 0\end{cases}$。

在多序列情形下，设序列的数目为 n，$D=\frac{1}{n}\sum_{j=1}^{n} d_j$，则有

$$\begin{aligned} \mathbf{W}&=\sum_{j=1}^{n}\mathbf{w}_j \qquad \mathbf{Y}=\sum_{j=1}^{n}\mathbf{y}_j \\ \mathbf{U}^{\alpha}&=\sum_{j=1}^{n}\mathbf{u}_j^{\alpha} \quad \mathbf{V}^{\alpha}=\sum_{j=1}^{n}\mathbf{v}_j^{\alpha} \end{aligned} \tag{2-44}$$

上述各矩阵都是对称矩阵。

在形式上，如果各分量所对应的属性彼此相互独立，则位置权重函数矩阵简化成 m 阶对角阵

$$\mathbf{I}(k)=\begin{bmatrix} I_{11}(k) & 0 & \cdots & 0 & \cdots & 0 \\ 0 & I_{22}(k) & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I_{\alpha=\beta}(k) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & I_{mm}(k) \end{bmatrix}_{m\times m}$$

特别地，若各位置平权，则该对角阵为常数对角阵，相当于一个常数 c 与单位阵的乘积。若再设该常数 $c=1$，则位置权重函数矩阵就变成了单位阵。

2.6.2 几何表示——对偶曲线

解析数论模型中的对偶描述子不仅仅是一组抽象的参数，它是可以直接拿给人看的。前面给出了标量形式和矢量形式的对偶公式。通过对偶公式所定义的对偶变量，对应于实空间中的点。若组成权重因子是 m 维实矢量，则相应的模式描述函数给出的就是一个 m 维实矢量的序列。对偶公式作为模式描述函数的前 l 项加权和，它所给出的仍是 m 维实矢量的序列。若规定 $l=0$ 时，$\mathbf{s}(l)=\mathbf{0}$，则该序列对应于 m 维实空间中过坐标原点的一条曲线，称为**对偶曲线**(D curve)。对偶曲线就是对偶变量的序列在实空间中的轨迹。在直观意义上，为了 D 曲线有一个跌宕起伏的外观，应注意以下原则。

原则 2 编码矩阵的每行的各元素之和应接近于 0，即

$$\sum_{i=1}^{n} x_{ji}=0 \quad (j=1,2,\cdots,m)。 \tag{2-45}$$

这一原则称为组成权重因子的**零均值原则**。该原则在小波变换理论中称作母小波函数的容许性条件，是为了保证逆小波变换能够实现[131]。这里的零均值原则也是为了把对偶曲线的振幅限定在一定的范围内（即使能量有限）。通过前面对偶描述子的学习而获得的组成权重因子可能不直接满足上面的零均值原则。此时，可以在如下两条性质的保证下，通过变换使其满足。

性质 1 编码矩阵的各列 $\mathbf{x}_i \quad (\mathbf{x}_i\in R^m; i=1,2,\cdots,n)$ 都加上同一个常矢量 $\mathbf{c}\in R^m$，变成 $\mathbf{x}'_i \quad (\mathbf{x}'_i\in R^m; i=1,2,\cdots,n)$，即

$$\mathbf{x}_i + \mathbf{c} = \mathbf{x}'_i \quad (\mathbf{x}_i, \mathbf{x}'_i, \mathbf{c} \in R^m; i = 1,2,\cdots,n) \tag{2-46}$$

则根据对偶公式，对各序列提取的对偶变量在 m 维实空间中的相对分布（可以用样本协方差矩阵的特征向量刻画）不变。

证明：将 $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{c}$ 代入对偶公式 $\mathbf{s}(l) = \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}[k] \quad (l = 1,2,\cdots,L)$ 有

$$\mathbf{s}(l) = \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}'[k] = \sum_{k=1}^{l} \mathbf{I}(k)(\mathbf{x}[k] + \mathbf{c}) = \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}[k] + \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{c}$$
$$= \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}[k] + \mathbf{s}_{\mathbf{c}}$$

式中 $\mathbf{s}_{\mathbf{c}}$ 为一个常矢量。由此可知，变换 $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{c}$ 的效果相当于将各样本所对应的对偶变量整体平移，故而不改变样本间的相对分布。

性质 2 编码矩阵的各列 $\mathbf{x}_i \quad (\mathbf{x}_i \in R^m; i = 1,2,\cdots,n)$ 都乘上同一个 非 0 常数 $c \in R - \{0\}$，变成 $\mathbf{x}''_i \quad (\mathbf{x}''_i \in R^m; i = 1,2,\cdots,n)$，即

$$c\mathbf{x}_i = \mathbf{x}''_i \quad (\mathbf{x}_i, \mathbf{x}''_i \in R^m, c \in R - \{0\}; i = 1,2,\cdots,n) \tag{2-47}$$

则根据对偶公式，对各序列提取的对偶变量在 m 维实空间中的相对分布不变。

证明：将 $\mathbf{x}''_i = c\mathbf{x}_i$ 代入对偶公式 $\mathbf{s}(l) = \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}[k] \quad (l = 1,2,\cdots,L)$ 有

$$\mathbf{s}(l) = \sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}''[k] = \sum_{k=1}^{l} \mathbf{I}(k)c\mathbf{x}[k] = c\left(\sum_{k=1}^{l} \mathbf{I}(k)\mathbf{x}[k]\right)$$

由于 c 是一个非 0 实常数，因此，变换 $\mathbf{x}''_i = c\mathbf{x}_i$ 的效果相当于将各样本所对应的对偶变量整体放缩，故而不改变样本间的相对分布。

上面的两条性质，实际上描述的是两种线性变换下，各样本对偶变量相对分布的性质。结论表明，用样本协方差矩阵的特征向量（或者是归一化处理后的样本协方差矩阵本身）刻画的对偶变量的相对分布，对于以上两种变换来说，是不变量。

下面看一个例子。在这个例子中，将画出大肠杆菌 *E. coli* K12 基因组的 D 曲线。为了简单起见，对偶描述子就取作标量形式，即一维 D 曲线。位置权重函数取为常函数 $I(k) = const = 1$，即假设各位置平权。设起初有如下映射：

A→1，C→2，G→3，T→4

此时的一维 D 曲线，如图 2-3 所示。

对以上映射施加如下变换

$$\begin{bmatrix} A \\ C \\ G \\ T \end{bmatrix} \to \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \times 2 - \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} = \begin{bmatrix} -3 \\ -1 \\ +1 \\ +3 \end{bmatrix}$$ 后成为新映射：A→-3，C→-1，G→+1，T→+3

变换后，满足原则 2。在该映射下，相应的 D 曲线为图 2-4。

图 2-3 变换前大肠杆菌 *E. coli* K12 基因组的一维 D 曲线

图 2-4 变换后大肠杆菌 *E. coli* K12 基因组的一维 D 曲线

由于以上的变换是线性变换，根据性质 1 和性质 2 可知，变换前后，不影响模式识别的聚类效果，却得到了跌宕起伏的图形。

等价表示定理

通过 $f:C\to X$（或 $C\to\mathbf{X}$）映射，把字符序列转化成数字（或实矢量）序列，再通过对偶公式转化成对偶变量的序列，乃至其几何形式高维 D 曲线。这样变来变去，会不会丢失信息，是一个值得讨论的问题。下面给出的“等价表示定理”，说明了什么情况下，这种表示不会丢失信息。

定理2.2 （**等价表示定理**）由集合 $\mathbf{X}=\left\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_i,\cdots,\mathbf{x}_n\mid\mathbf{x}_i\in R^m\right\}$ 中的元素组成的长度为 L 的 m 维实矢量的序列 $\mathbf{x}=[\mathbf{x}[1],\mathbf{x}[2],\cdots,\mathbf{x}[k],\cdots,\mathbf{x}[L]],\ \ (\mathbf{x}[k]\in\mathbf{X})$，与其前 l 项带权和序列 $\mathbf{s}=[\mathbf{s}[1],\mathbf{s}[2],\cdots,\mathbf{s}[l],\cdots,\mathbf{s}[L]],\ \ (\mathbf{s}[l]\in R^m)$，其中，通项由下面的“带权求和公式”给出，即 $\mathbf{s}(l)=\sum_{k=1}^{l}\mathbf{I}(k)\mathbf{x}[k]\ \ (l=1,2,\cdots,L)$，在位置权重函数矩阵 $\mathbf{I}(k)$ 满足 $\left|\mathbf{I}(k)\right|\neq 0,\forall k\in[1,L]$ 时，构成一一对应关系。

证明：将“带权和序列”的通项公式 $s(l)=\sum_{k=1}^{l}\mathbf{I}(k)\mathbf{x}[k]\ \ (l=1,2,\cdots,L)$ 写成如下递推形式

$$\begin{cases}\mathbf{s}[l+1]=\mathbf{s}[l]+\mathbf{I}(l+1)\mathbf{x}[l+1]\ \ (l=0,1,2,\cdots,L-1)\\ \mathbf{s}[0]=\mathbf{0}\end{cases}\tag{2-48}$$

对于一个给定的矢量序列 $\mathbf{x}$，考虑从左到右的递推求和过程，即 l 从 0 变到 $L-1$ 的过程。当取定一个 $l=l_0\in\{0,1,2,\cdots,L-1\}$ 时，由（2-48）式的递推关系得

$$\mathbf{s}[l_0+1]=\mathbf{s}[l_0]+\mathbf{I}(l_0+1)\mathbf{x}[l_0+1]\tag{2-49}$$

对于确定的位置 l_0+1 而言，$\mathbf{s}[l_0]$ 是前 l_0 步递推求和的结果，即为已经得出的常矢量，设其为 $\mathbf{s}_{l_0}$；又因为，$\mathbf{I}(k)$ 满足 $\left|\mathbf{I}(k)\right|\neq 0,\forall k\in[1,L]$，故 $\mathbf{I}(k)\neq\mathbf{0}$，即，$\mathbf{I}(l_0+1)$ 对于取定的位置 l_0+1 而言，是确定的“非零实矩阵”，设其为 $\mathbf{I}_{l_0+1}$，即 $\mathbf{I}(l_0+1)=\mathbf{I}_{l_0+1}\neq\mathbf{0}$；因此，（2-49）式可写成，

$$\mathbf{s}[l_0+1]=\mathbf{s}[l_0]+\mathbf{I}(l_0+1)\mathbf{x}[l_0+1]=\mathbf{s}_{l_0}+\mathbf{I}_{l_0+1}\mathbf{x}[l_0+1]\tag{2-50}$$

由（2-50）式可以看出，$\mathbf{s}[l_0+1]$ 是 $\mathbf{x}[l_0+1]$ 的线性函数，即由 $\mathbf{x}[l_0+1]$ 的取值可以唯一

地确定$\mathbf{s}[l_0+1]$的取值。因为，对于$\forall l_0 \in \{0,1,2,\cdots,L-1\}$，（2-50）式都成立，所以，矢量序列$\mathbf{x}$可以唯一地确定其“前$l$项带权和”的序列$\mathbf{s}$。

反过来，若已知某m维实矢量的序列$\mathbf{x}$的“前l项带权和”序列$\mathbf{s}$，即满足条件

$\mathbf{I}^{-1}(l+1)(\mathbf{s}[l+1]-\mathbf{s}[l]) \in \mathbf{X} \subset R^m \quad \forall l \in \{0,1,2,\cdots,L-1\}$的序列$\mathbf{s}$，因为，位置权重函数矩阵$\mathbf{I}(k)$满足$|\mathbf{I}(k)| \neq 0, \forall k \in [1,L]$，所以，由（2-48）式中的递推关系反解得到

$$\mathbf{x}[l+1] = \mathbf{I}^{-1}(l+1)(\mathbf{s}[l+1]-\mathbf{s}[l]) \quad (l=0,1,2,\cdots,L-1) \tag{2-51}$$

对于任意给定的$l=l_0 \in \{0,1,2,\cdots,L-1\}$，有

$$\mathbf{x}[l_0+1] = \mathbf{I}^{-1}(l_0+1)(\mathbf{s}[l_0+1]-\mathbf{s}[l_0]) = \mathbf{I}_{l_0+1}^{-1}(\mathbf{s}_{l_0+1}-\mathbf{s}_{l_0}) \tag{2-52}$$

因为，$\mathbf{I}_{l_0+1}^{-1}$是确定的“非零常矩阵”，所以，对于任意给定的$l_0 \in \{0,1,2,\cdots,L-1\}$，$\mathbf{x}[l_0+1]$都可以通过(2-52)式，由序列$\mathbf{s}$中，相邻两项$\mathbf{s}_{l_0+1}$与$\mathbf{s}_{l_0}$的差唯一地确定。因此，矢量序列$\mathbf{x}$可由其前$l$项和序列$\mathbf{s}$唯一地确定。

综上两方面，则有结论：在位置权重函数矩阵$\mathbf{I}(k)$满足$|\mathbf{I}(k)| \neq 0, \forall k \in [1,L]$的情况下，矢量序列$\mathbf{x}$与其“前$l$项带权和”序列$\mathbf{s}$之间存在一一对应关系，即是说，矢量序列$\mathbf{x}$ 和它的“前l项带权和”序列$\mathbf{s}$在彼此可重构的意义下互为等价表示。至此“**等价表示定理**”证明完毕。

根据等价性的递推关系（若$A \Leftrightarrow B, B \Leftrightarrow C$,则$A \Leftrightarrow C$）可知，只要$f: C \to \mathbf{X}$是“一对一”映射，并且位置权重函数矩阵满足$|\mathbf{I}(k)| \neq 0, \forall k \in [1,L]$，就会有如下等价关系成立：

$$s \Leftrightarrow \mathbf{x} \Leftrightarrow \mathbf{s} \Leftrightarrow \text{D curve}, \tag{2-53}$$

其中，s表示字符序列，$\mathbf{x}$表示实矢量序列，$\mathbf{s}$表示对偶矢量序列，D curve是它的几何表示。

s到$\mathbf{x}$的转化是直接的，即通过$f: C \to \mathbf{X}$映射直接对应；$\mathbf{s}$到D curve的转化也是直接的代数表示与几何表示的关系；而“等价表示定理”打通了中间环节$\mathbf{x} \Leftrightarrow \mathbf{s}$。这样一来，什么条件下，没有信息损失，就比较清楚了。标量形式作为矢量形式的特例（1维矢量）自动满足以上关系。

2.6.3 应用扩展

解析数论模型的核心概念是对偶描述子。在把对偶描述子用于字符序列的特征提取和分类识别时，还可以进一步扩展其形式。例如，考虑字符序列中，相邻字符之间的关联，而有高阶对偶描述子，多个对偶描述子联合使用来描述字符序列的多对偶描述子等形式。

高阶对偶描述子 现在考虑由字符集C中的双字母组合所构成的集合，即C的二次直幂$C\times C=C^2$，式中“×”表示集合的笛卡尔积。由于C中有n个元素（字符），C^2中就有n^2个元素，这些元素都是长度为2的字符串，因此$C^2\subset C^*$。这n^2个元素可以看作n^2个新字符，**二阶对偶描述子**就是指由C^2中的元素所对应的组成权重因子和引入的相应的位置权重函数所构成的对偶描述子。一般地，由字符集C中的 n 字母组合所构成的集合，即C的 n 次直幂$\overset{n}{\underset{i=1}{\times}}C_i=C^n$，可以定义 **n 阶对偶描述子**。高阶对偶描述子的模式描述函数和模式偏离函数可以和一阶对偶描述子一样去定义。由于标准模式的模式描述函数是常数，故而对于各阶对偶描述子来说，它可以是不变的，即恒为同一个常数，通常取作1。

用高阶对偶描述子描述字符序列时，依赖于子序列的取法，有两种描述方式：线性描述和非线性描述。下面举例说明。

设有一长度为$L_0=48$的DNA序列如下，

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAA

如果用三阶对偶描述子描述，即每次取该序列的三个字符的子串，则第一个子串为“AGC”，第二个子串有两种取法，分别对应于两种描述方式：

（1） **线性描述**。线性描述采用的是一种称作“滑动窗口”的技术。从左向右，每次滑动一个字符，即第二个子串为“GCT”，第三个子串为“CTT”，……以此类推。线性描述具有最大的信息冗余度。

（2） **非线性描述**。以一种拼接（或叫覆盖 tiling）的方式，取过的字符就不再取，即每隔三个取一个，第二个子串为“TTT”，第三个子串为“TCA”，……以此类推。这种非线性描述没有重复使用原序列中的字符，因而描述无冗余。

高阶对偶描述子考虑了字符序列的局部关联，增加了组成权重因子的个数。对于n阶对偶描述子来说，它的组成权重因子的个数为字符集元素个数的n次幂。由于高阶对偶描述子的组成权重因子实际上是原序列的子串，因此，导致了“线性”与“非线性”两种描述方式

的出现。至于为何这么叫，在紧接下来的“立体描述”部分说明。

立体描述：基于“自相似”的想法，递归运用对偶描述子，可以给出字符序列的一个从低到高的整体的描述。先给出对偶公式的递归运用形式：

$$\begin{cases} S^n(l^n)=\sum_{k_l^n=1}^{l^n} I^n(k_l^n)\times S^{n-1}(l^{n-1}) \quad (n=1,2,\cdots,\left\lfloor \frac{L_0-l}{l-1}+1 \right\rfloor \text{ or } \left\lfloor \log_{\frac{1}{l}} \frac{l}{L_0}+1 \right\rfloor \\ S^0(l)=\sum_{k=1}^{l} I(k)\times x[k] \end{cases} \tag{2-54}$$

式中上标 n 表示递归过程中的第 n 层, l 表示每次考察长度为 l 的子串， k_l^n 为该子串首字符在第 n 层序列中出现的位置； l^n 表示第 n 层序列中以长度为 l 的子串度量的序列的前 l 项。 $\lfloor \square \rfloor$ 表示不大于“□”的最大整数。究竟是选 $\left\lfloor \frac{L_0-l}{l-1}+1 \right\rfloor$ 还是 $\left\lfloor \log_{\frac{1}{l}} \frac{l}{L_0}+1 \right\rfloor$，取决于描述方式。

如果用线性描述方式，从左向右完成一遍扫描后（即 n=1 时），序列长度由原来的 L_0 缩短成 L_0-l+1 。递归运用上述公式，完成第二遍扫描后(n=2)，序列长度又缩短为 L_0-2l+2，……以此类推。序列长度每次缩短 $l-1$，递缩的公式为，

$$L_{n+1}=L_n-(l-1) \quad (n=0,1,2,\cdots,\left\lfloor \frac{L_0-l}{l-1}+1 \right\rfloor) \tag{2-55}$$

写成非递推形式，

$$L_n=L_0-n(l-1) \quad (n=0,1,2,\cdots,\left\lfloor \frac{L_0-l}{l-1}+1 \right\rfloor) \tag{2-56}$$

从上式可见， L_n 为 n 的线性函数，故称这种描述方式为“线性描述”。

如果采用非线性描述，则序列长度的递缩公式为

$$L_{n+1}=\frac{L_n}{l} \quad (n=0,1,2,\cdots,\left\lfloor \log_{\frac{1}{l}} \frac{l}{L_0}+1 \right\rfloor) \tag{2-57}$$

写成非递推形式，

$$L_n=(\frac{1}{l})^n L_0 \quad (n=0,1,2,\cdots,\left\lfloor \log_{\frac{1}{l}} \frac{l}{L_0}+1 \right\rfloor) \tag{2-58}$$

可以看出， L_n 是 n 的非线性函数，确切地说，序列是呈指数递缩的。这时，采用的不

是“滑动窗口技术”，就是以每l长度为单位，从左向右依次扫描。

事实上，公式（2-57）和（2-58）是有问题的，即不能保证每次缩短后，序列长度L_n都能被l整除，即使原序列s的长度L_0能被l整除的话。为了克服这一问题，在递缩过程中，可以在不足l整数倍的序列右端补 0(zero padding)，或者干脆把多余的右端扔掉，再或者，用一种更为精巧的办法——“对称追补”(symmetric padding)，即尾端缺少几个，就从尾端开始向前找几个，将找到的子串，翻转 180 度后，追加到尾端，以便使递缩可以进行下去。这样每缩短一次，统计一遍，每统计一遍，就可以得到一组特征量；递缩到最后，整个序列总能 “缩编”成一个数字，该数字也可作为原字符序列s的特征量。

像上面这样递归运用对偶公式，每“上升”一层，序列的长度就缩短一些，使描述行为体现出类似“金字塔”式的“立体结构”，故而称这种描述方法为“立体描述”。

多对偶描述子 多对偶描述子是为描述序列特征而联合使用的一组对偶描述子。这些描述子可以是同阶的，也可以是不同阶的。其中，每一个描述子描述了序列的一部分特征，合起来给出序列的一个较完整的描述。

2.7 Z 曲线理论简介

Z曲线理论是张春霆教授提出的DNA序列的一种等价的几何表示形式[48,49]。近年来，Z曲线理论以其简单、直观的特点，在DNA序列分析中得到了广泛的应用，并受到学术界越来越多的关注[50-53]。下面从对偶描述子理论的角度看一下Z曲线。对于4种核苷酸构成的字符集$C=\{a,c,g,t\}$，考察核苷酸的三种属性：（1）核苷酸类型（嘌呤或嘧啶）；（2）环中对应位置上的基团性质（氨基或酮基）；（3）形成DNA双链时氢键的数目或强弱（强氢或弱氢），并把它们量化（形式上的），就得到如下常矢量的集合$\mathbf{X}=\left\{\mathbf{x}_a,\mathbf{x}_c,\mathbf{x}_g,\mathbf{x}_t\right\}\ (\mathbf{x}_i\in R^3)$，其中$\mathbf{x}_a=[+1,+1,+1]$，$\mathbf{x}_c=[-1,+1,-1]$，$\mathbf{x}_g=[+1,-1,-1]$，$\mathbf{x}_t=[-1,-1,+1]$。这里，$C\to\mathbf{X}$映射是一个“一对一”的映射，即$a\to\mathbf{x}_a, c\to\mathbf{x}_c, g\to\mathbf{x}_g, t\to\mathbf{x}_t$。把这4个3维矢量写成如下编码矩阵

$$\begin{array}{c} \\ x \\ y \\ z \end{array}\begin{array}{c} \begin{array}{cccc} a & c & g & t \end{array} \\ \begin{bmatrix} +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} \end{array} \tag{2-59}$$

该矩阵为行正交矩阵，即满足前面的“原则1”，因而表示无冗余；又，每行的各元素之和为0，即满足前面的“原则2”，因而，有跌宕起伏的几何外观。

在Z曲线理论中，位置权重函数矩阵为3阶单位阵，即

$$\mathbf{I}(k)=const=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \tag{2-60}$$

因为，单位矩阵满足条件$\left|\mathbf{I}(k)\right|\neq 0,\forall k\in[1,L]$，因而，由上节的“等价表示定理”可知，3维Z曲线构成原字符序列的等价几何表示。又由于编码矩阵为行正交矩阵，故而，该3维表示无冗余。

一个有趣的问题是，用Z曲线的一个分量或两个分量能否构成原字符序列的等价表示呢？先看单分量的情形。拿x分量为例，由于映射$C\to\mathbf{X}_x$为$a\to+1,c\to-1,g\to+1,t\to-1$，即$a,g$都对应于$+1$，$c,t$都对应于$-1$，不是“一对一”映射，因而，在$x$分量中便无法区分$a$与$g$，或$c$与$t$。因此，仅$x$分量无法构成原字符序列的等价表示。同样，只用$y$分量或$z$分量，也无法构成原序列的等价表示。

那么，用两个分量能否构成原字符序列的等价表示呢？答案是肯定的。因为，无论是$C\to\mathbf{X}_{xy}$、$C\to\mathbf{X}_{yz}$或是$C\to\mathbf{X}_{zx}$都是“一对一”的映射。例如，$C\to\mathbf{X}_{yz}$映射为$a\to[+1,+1],c\to[+1,-1],g\to[-1,-1],t\to[-1,+1]$，$a,c,g,t$分别对应于不同的2维常矢量，是“一对一”的映射。同样，$C\to\mathbf{X}_{xy}$和$C\to\mathbf{X}_{zx}$也都是“一对一”的映射。而此时的2阶单位阵，也满足条件$\left|\mathbf{I}(k)\right|\neq 0,\forall k\in[1,L]$，因此，Z曲线的任意两个分量，都可以构成原字符序列的等价表示。又由于“编码矩阵”(2-74)的行正交关系，所以，这些2维表示也都是无冗余的。

这里的2维是无冗余，前面的3维也是无冗余，那么，关于表示的冗余性，有没有一般性的结论呢？有。

结论1 （冗余性定理）对于n字符集合中的字符组成的一维字符序列，在位置权重函数矩阵满足$\left|\mathbf{I}(k)\right|\neq 0,\forall k\in[1,L]$的情况下，由对偶公式得到的对偶变量的序列，若要求它是

原字符串的无冗余表示，则这种表示最多到n维。

证明：对于m行n列的“编码矩阵”，即

$$M_{m\times n}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1i} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2i} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{j1} & x_{j2} & \cdots & x_{ji} & \cdots & x_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mi} & \cdots & x_{mn} \end{bmatrix} \quad (x_{ji}\in R-\{0\}; i=1,2,\cdots,n, j=1,2,\cdots,m)$$

由线性代数的知识可知：当$m>n$时，如果前n个行向量彼此正交，则它们必线性无关，从而构成n维线性空间的一组基底，那么，第$n+1$个行向量，必可表示成前n个行向量的线性组合，因而，不可能再与它们正交[132]。所以，当$m>n$时，“编码矩阵”不可能是（由非零行向量构成的）行正交矩阵。故而，由对偶变量序列构成的原字符串的无冗余表示最多到n维（n为字符集中元素的个数）。

当然，表示的无冗余性，并非必然要求。有时带些冗余，用起来会更方便一些。

Z曲线用于基因识别等领域时，考虑密码子不同相位的特异性，又将变量推广到9维空间，此时，

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \end{bmatrix} = \sum_{k=1}^{L} \begin{bmatrix} I_1(k) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_1(k) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_1(k) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2(k) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_2(k) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_2(k) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_3(k) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_3(k) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_3(k) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ x_3 \\ y_3 \\ z_3 \end{bmatrix}$$

其中，x_1, y_1, z_1，x_2, y_2, z_2，x_3, y_3, z_3分别对应于3个相位，相应的位置权重函数为

$$I_1(k)=\begin{cases}1 & \text{if } k \bmod 3=1 \\ 0 & \text{else}\end{cases}, \quad I_2(k)=\begin{cases}1 & \text{if } k \bmod 3=2 \\ 0 & \text{else}\end{cases}, \quad I_3(k)=\begin{cases}1 & \text{if } k \bmod 3=0 \\ 0 & \text{else}\end{cases}。$$

上面的公式，只是形式化的写法，并不真正符合对偶公式的要求。一方面，按相位提取

的Z曲线，并不是真正意义上的9维曲线（没有给出字符集到9维常矢量集的映射），而是将原字符序列按相位拆分成了三条子序列，再分别提取Z曲线，因此，实际上是3条3维曲线，而非1条9维曲线。另一方面，上面写出的位置权重函数矩阵，在每一个位置 k 处，均不满秩，因此，不符合条件 $\left|\mathbf{I}(k)\right| \neq 0, \forall k \in [1, L]$。正因为如此，上面的公式所给出的9参数序列，也可以是原字符序列的无冗余的等价表示。

Z曲线在考虑了相邻碱基的关联后，又推广了的多参数形式，则可以用高阶对偶描述子描述之。

上面，虽然在代数形式上，用对偶公式给出了Z曲线理论的数学描述，但却并不是真正意义上的Z曲线理论的发展或推广。Z曲线理论的精髓，在于其深刻的对称性的思想和几何形式上的“形而上的”优美。而这些，是作者未能领会的。因此，上面公式对Z曲线理论的描述，也只是形式上的。

2.8 基于位置权重的序列分析方法之——“位置权重矩阵”

出于逻辑上的完整性，本节介绍基于位置权重的序列分析方法的另一分支：位置权重矩阵，并指出它与对偶描述子方法的内在联系。

在 2.2 节中，将模式描述函数写成

$$N(k) = I(k) \times x[k] \quad (k = 1, 2, \cdots, L) \tag{2-5}$$

式中，将 $N(k)$ 写成 $I(k)$ 与 $x[k]$ 两项的乘积，是假定某位置 k 处的权重，仅与位置 k 有关，而与该位置处的内容（即出现的是哪个字符），没有关系。因此，上面的分离变量形式的公式是模式描述函数的内容无关（content free）形式。一般情况下，可能是内容相关(content sensitive)的，那么 $N(k)$ 可以写成

$$N(k) = f(k, x[k]) \tag{2-61}$$

此时，可用一个二维数表 T 来表示模式描述函数，即

	1	2	⋯	k	⋯	L
x_1	T[x_1][1]	T[x_1][2]	⋯	T[x_1][k]	⋯	T[x_1][L]
⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_i	T[x_i][1]	T[x_i][2]	⋯	T[x_i][k]	⋯	T[x_i][L]
⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_n	T[x_n][1]	T[x_n][2]	⋯	T[x_n][k]	⋯	T[x_n][L]

该二维数表通常称作“位置权重矩阵”，它的元素 T[x_i][k]通常是通过统计多条等长序列中每一个位置k处，各字符出现的频率而得到的[72]。设序列的数目为n，则有

$$\mathrm{T}[x_i][k]=\frac{n_{x_i}[k]}{n}\quad(k=1,2,\cdots,L) \tag{2-62}$$

式中$n_{x_i}[k]$表示位置k出现字符c_i的次数，因此有$\sum_{x_i\in X} n_{x_i}[k]=n\quad(k=1,2,\cdots,L)$。

位置权重矩阵适合于处理长度一致的多条字符序列，它有一种可视化表示方法，称为**Logo 图**[133]。这种图形，可以根据不同位置处，各字符出现的频率大小，选取不同的字高，来显示该字符，从而非常直观地显示出序列中各位置的保守程度[134]。

借助于二维数表 T，模式描述函数可以写成

$$N(k)=\mathrm{T}[\sum_{x_i\in X} x_i\delta(x[k]-x_i)][k]\quad(x[k]\in X;\ k=1,2,\cdots,L) \tag{2-63}$$

式中，$\delta(x[k]-x_i)=\begin{cases}1 & \text{if } x[k]=x_i\\ 0 & \text{if } x[k]\neq x_i\end{cases}$ (2-64)

而对偶公式为

$$S(l)=\sum_{k=1}^{l}N(k)=\sum_{k=1}^{l}\mathrm{T}[\sum_{x_i\in X} x_i\delta(x[k]-x_i)][k] \tag{2-65}$$

在实际应用中，通常取序列中某一点附近左右长各为l和l'的子序列去做统计。因此，用对偶公式对该子序列打分，即为对偶变量的值

$$\begin{aligned}&S([x[-l]\cdots x[k-1]x[k]x[k+1]\cdots x[l']])=\sum_{k=-l}^{l'}f(k,x[k])\\&=\sum_{k=-l}^{l'}\mathrm{T}[\sum_{x_i\in X} x_i\delta(x[k]-x_i)][k]\end{aligned} \tag{2-66}$$

其中，l前面的“$-$”表示字符出现在该点的上游（左边）。

这里，用对偶公式对子序列打分的方法，和传统的位置权重矩阵的打分方法，略有不同。传统上的位置权重矩阵基于条件概率进行打分，用的是“连乘”的形式，即

$$\begin{aligned}&Score([x[-l]\cdots x[k-1]x[k]x[k+1]\cdots x[l'])=\prod_{k=-l}^{l'}f(k,x[k])\\&=\prod_{k=-l}^{l'}\mathrm{T}[\sum_{x_i\in X} x_i\delta(x[k]-x_i)][k]\end{aligned} \tag{2-67}$$

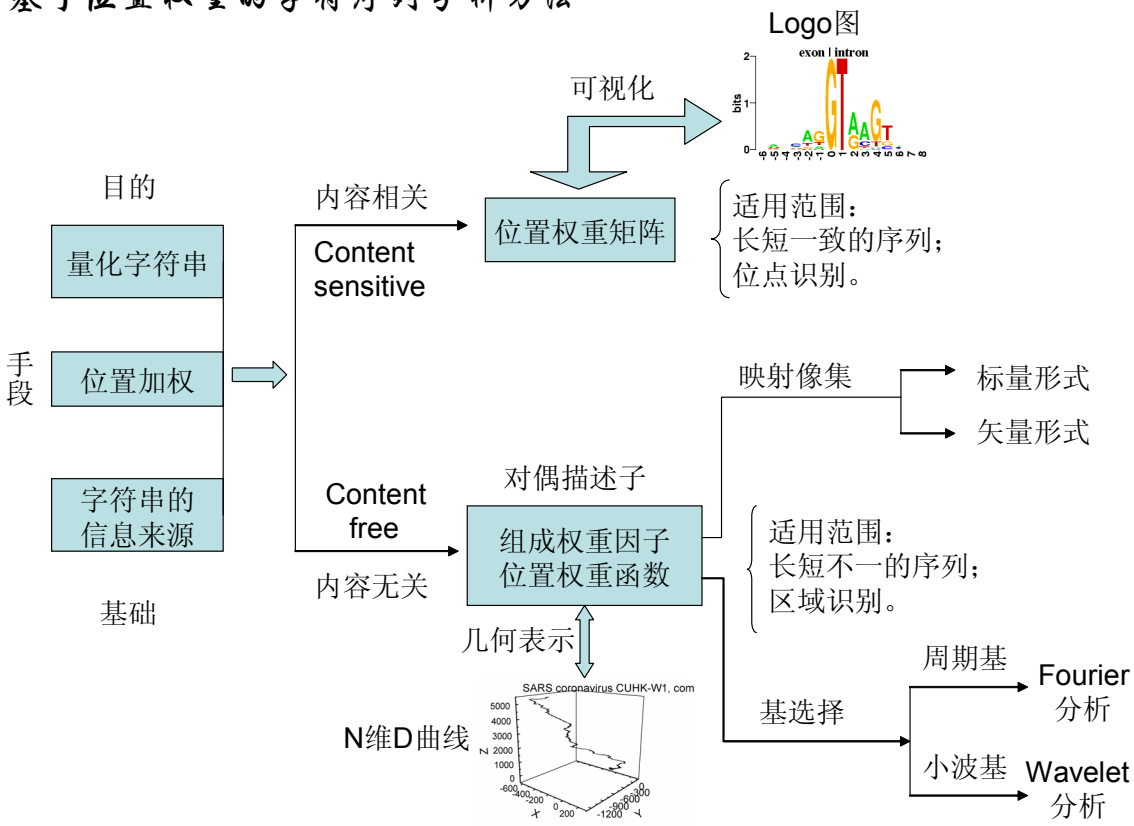
或取对数后，转化成“连加”的形式

$$Score=\log\left(\prod_{k=-l}^{l'}f(k,x[k])\right)=\sum_{k=-l}^{l'}\log f(k,x[k])$$
$$=\sum_{k=-l}^{l'}\log\mathrm{T}[\sum_{x_i\in X}x_i\delta(x[k]-x_i)][k] \quad , \qquad (2\text{-}68)$$

而用对偶公式进行打分时，直接使用连加(2-66)。

本章小结

下面，把本章的内容用下图概括。从图中，可以更清楚地看出知识的脉络层次。



由上图可以看出：奔着“量化字符串”的目的，基于“字符串信息来源”的分析，通过“位置加权”的手段，而衍生出来的字符序列分析方法，分为两个分支：“位置权重矩阵”和“对偶描述子”。位置权重矩阵适合于处理长度一致的字符序列，通常用于位点的识别，如酶剪切位点。对偶描述子可以处理长短不一的字符序列，适合于区域识别，如蛋白质编码区的识别。它们互为补充，形成了基于位置权重的字符序列分析方法体系。在对偶描述子方法中，依据组成权重因子的性质（标量还是矢量），即映射像集的选择，分别对应于标量形式和矢量形式；依据位置权重函数展开基的选择，分别关联到傅立叶分析和小波分析等内容。

这两种方法都有相应的可视化表示形式：位置权重矩阵对应于 Logo 图，而对偶描述子对应于多维 D 曲线。本章讨论了字符序列数量化表示的等价性问题（与位置权重函数矩阵相关），给出了“等价表示定理”；还讨论了字符序列数量化表示的冗余性问题（与组成权重因子的选择相关），给出了无冗余表示的最高维数的结论。

另外，当考虑字符序列内的局部关联（多字母组合）时，引入了高阶描述子的概念，高阶描述子的使用必然带来“线性描述”与“非线性描述”两种描述方式的出现。基于“自相似”的想法，给出了对偶公式嵌套运用的形式，从而导致了“立体描述”的出现。

字符序列解析数论模型的思路简明而自然。它把字符序列看成是数的表示（在自然科学中，数本来就是用字符序列表示的），从而把字符序列分析问题转化成一个数论问题。为了逼近“位权函数”，通过展开基的选择，自然地将傅立叶分析、小波分析等内容引入字符序列的研究，即，应用分析手段来处理数论问题，这就是字符序列解析数论模型（Analytic Number Theory Model）的含义。字符序列解析数论模型的核心概念是对偶描述子。对偶描述子不是一个凭空捏造的概念，有着确切的内涵和外延。对偶描述子的两个组成部分：组成权重因子和位置权重函数，直接来源于自然计数制中基数和位权函数的概念，是它们在实数域上的推广。下面解释一下这个名称含义。“对偶”，是指组成权重因子和位置权重函数之间的相互依赖、此消彼长的关系。“描述”，强调它是从数论的角度给出的字符序列的一个描述，或叫模型，而且，用它进行模式特征的提取也是基于样本描述的。“子”，是由于组成权重因子和位置权重函数之间相互依赖、密不可分，拆开来，单看其中的一个，没有意义，因此，将它们打成一个包，以便数说，例如，一个描述子、两个描述子，一阶描述子，高阶描述子等。

第三章 对偶描述子方法在生物信息学中的应用举例

字符序列的解析数论模型，亦即对偶描述子方法，是从数论的角度看待字符序列分析问题，把扰动法的思想引入统计领域，以位置加权频率来反映字符序列的组成和排列两方面的信息，是处理字符序列问题的一般性方法。它本身并不限定应用领域，只要是涉及字符序列分析的问题，都可以尝试应用该方法解决。但近年来，由于测序工作的开展，生物信息学领域中产生了大量的生物序列数据，恰好为该方法提供了用武之地。另一方面，该模型最初也正是应此需要而提出的。

本章举了一些生物信息学领域中字符序列分析的例子，旨在演示字符序列的解析数论模型在字符序列分析中的应用，而并不刻意于解决某个问题，因此，所得的结果均有进一步提高的余地，并不代表该方法所能取得的最好水平。这些例子主要涉及 DNA 序列分析领域，如：冠状病毒基因组序列的特征提取、原核生物蛋白质编码区的特征提取与识别、人类基因组外显子的识别等。本拟再举一些在蛋白质二级结构预测中应用的例子，但由于个人的时间和精力有限，仓促完稿，只好作罢了。

3.1 对偶描述子用于字符序列特征提取——对偶描述子的学习演示

对偶描述子用于字符序列特征提取的理论基础是**函数逼近理论**。它将位置权重函数在一组基上作展开，并用展开式的系数来表示序列的特征。在确定系数的方法上，比较独特，不同于通常的分析方法。在通常的分析方法中，如傅里叶变换、小波变换乃至 K—L 变换等，系数的确定是通过计算原始信号与基函数的内积来实现的。而在对偶描述子理论中，则是通过相对于某一参考点偏离极小的条件，通过解线性方程组得到的。本节演示了对偶描述子用于字符序列特征提取的过程，即对偶描述子的学习过程。第一小节演示了对偶描述子的一次性学习与字符序列的重构。第二小节演示了对偶描述子的交替式学习过程。在交替式学习的情况下，表征字符序列特征的参数除了位置权重函数的展开式系数以外，还包括了与它对应的组成权重因子，即整个对偶描述子。第三小节，作为高阶对偶描述子的例子，演示了二阶对偶描述子的交替式学习过程，着重指出了两种描述方式对学习结果的影响。

3.1.1 冠状病毒基因组序列的特征提取

——对偶描述子的一次性学习演示

材料与方法

材料 用的是香港大学测定的冠状病毒基因组 SARS coronavirus HKU-39849，该基因组共有 29742 个碱基，G+C 含量为 40.76。它是从 NCBI 下载的。

方法 使用映射 $f:C \to X = \begin{matrix}\{\\ \updownarrow \\ \{\end{matrix}\begin{matrix}A & C & G & T\\ \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 1 & 2 & 3 & 4\end{matrix}\begin{matrix}\}\\ \\ \}\end{matrix}$ 将基因组字符序列转化成实数的序列，再用第二章的公式（2-21 一般展开式）逼近位置权重函数，取余弦三角基，则有：

$$I(k)=\sum_{m=1}^{n} a_m \cos\frac{2\pi k}{m} \quad (n=4) \tag{3-1}$$

这里，取 $n=4$ 表示取展开式的前 4 项，相当于考虑 1-4 的周期性。

应用第二章中的（2-13）求出展开式中的系数向量 **a**，然后再由位置权重函数根据(2-17)式重构出 SARS 基因组的字符序列，并根据（2-18）和（2-19）式计算了重构的失真率。

结果与讨论

结果 将所求得的结果列于表 3-1 中。

表 3-1 对偶描述子用于冠状病毒基因组的特征提取

物种	基因组大小	GC%	a_1	a_2	a_3	a_4	失真率 e	有效字符数
SARS HKU-39849	29742	40.76	75476	-40	-1352	68	0.715	1041

讨论 失真率 e=0.715 表示重构出来的基因组字符序列有 71.5%是不对的。若随机地从 4 字符集{A,C,G,T}中拿出字符形成 DNA 序列，则选对的概率为 1/4，即失真率为 75%。因此，相比较可得，被准确重构出来的字符数（有效字符数）为 29742 x (75%-71.5%)=1041。即用 4 个浮点数有效地表示了 1041 个字符。设在计算机中，每个浮点数用 4 个字节编码，每个字符用 1 个字节编码，则相当于用 4 x 4=16 个字节，有效地描述了 1041 个字节，压缩比为：1041∶16≈65∶1。

3.1.2 原核基因编码区公共特征的提取

——对偶描述子的交替式学习演示

原核生物的蛋白质编码区是一个完整连续的 ORF，其中，基本上不含有非编码序列。提取原核基因组蛋白质编码区的公共特征就是使用一个对偶描述子来描述一组蛋白质编码序列。这个对偶描述子就是这组序列公用的一个对偶描述子，携带了它们整体的信息，在一

定的程度上做了平均。

材料与方法

材料 这里使用大肠杆菌的K 12株，即 *Escherichia coli* K12，其基因组大小为4629221bp，G+C 的含量为 50.79%，其中含有 4289 个注释基因。作为对比，另外用到了物种新月柄杆菌 *Caulobacter crescentus* 的基因组，其大小为 4016947bp ，G+C 含量为 67.21%，共有 3737 个注释基因。它们是从 GenBank Release 131.0 数据库中下载的。

方法 提取这 4289 个注释基因的公共特征，是通过计算它们的模式偏离函数的总和 D 实现的。在第二章提到，使用对偶描述子的交替式学习方式，要在学习过程中避免奇异模式的出现。因此，若用三角余弦基逼近位权函数的话，最多从三角多项式的第二项开始：

$$I(k)=\sum_{m=2}^{n}a_m\cos\frac{2\pi k}{m}\quad(n=5) \tag{3-2}$$

这里还是取 4 项，即是说，考虑了序列中存在的 2、3、4 和 5 周期性。随机地给对偶描述子的参数：组成权重因子和位置权重函数展开式中的系数赋初值，然后，用第二章中的公式（2-14）和（2-15）、（2-16）及（2-33）交替求得对偶描述子参数的修正之后的值。在此学习过程中总的模式偏离函数的值 D 逐渐减小，如此进行下去，直至 D 值不再减小为止。此时所获得的组成权重因子和位置权重函数展开式系数的取值即为极佳对偶描述子的参数值。

为了观察不同的基函数选取方式，对最终的模式偏离函数极小值和极佳对偶描述子以及学习过程的影响，又选用了如下的基：

$$I(k)=\sum_{m=4}^{n}a_m\cos\frac{2\pi k}{m}\quad(n=7) \tag{3-3}$$

即是说，考虑了序列中存在的 4、5、6 和 7 周期性。又为了观察不同的序列特征对最终的模式偏离函数极小值和极佳对偶描述子以及学习过程的影响，还选择了物种 *Caulobacter crescentus* 的基因组作为对比。

结果与讨论

结果 这里总共给出了八个学习样例，分列于表 3-2、表 3-3、表 3-4 和表 3-5 等 4 个表格中。表格中列出了组成权重因子或（和）位置权重函数的初始值、总模式偏离函数的初始值、极佳对偶描述子的参数值、最终获得的 D 值以及学习过程中交替的次数。表 3-2 中列出的组成权重因子的初始值表示迭代过程是从预先给定的一组组成权重因子开始的；表 3-3 中列出位置权重函数展开式系数的初始值表示迭代过程是从预先给定的一个位置权重函数

开始的。而表 3-4 中列出的两个学习样例 5 和样例 6，则分别是从预先给定的组成权重因子和预先给定的位置权重函数开始学习过程的，但它们选用的是（3-3）所给出的余弦三角基。表 3-5 中列出的两个学习样例 7 和 8 用的是（3-2）式的余弦基，但所用的基因组是物种 *Caulobacter crescentus* 的。

表 3-2 描述 *Escherichia coli* K12 基因组的学习样例，列出了：组成权重因子的初始值、总模式偏离函数的初始值、极佳对偶描述子的参数值、最终获得的 D 值以及学习过程中交替的次数 [a]

学习样例 1	学习样例 2
组成权重因子初始值	组成权重因子初始值
x_a=1.0000000000	x_a=0.0003000000
x_c=2.0000000000	x_c=26999.0000000000
x_g=3.0000000000	x_g=-411.0000000000
x_t=4.0000000000	x_t=7.8200000000
总模式偏离函数的初始值	总模式偏离函数的初始值
D_0=117.0565169985	D_0=2702175464.4786720000
经过 5 次学习后，得到的极佳描述子	经过 2 次学习后，得到的极佳描述子
a_1=-0.0001771210	a_1=-0.0000000091
a_2=-0.0142719856	a_2=-0.0000007305
a_3=0.0000535288	a_3=0.0000000028
a_4=0.0001440630	a_4=0.0000000074
x_a=-3.3927708843	x_a=-66282.1693310537
x_c=1.0413107599	x_c=20343.7958709717
x_g=-16.7349123563	x_g=-326943.8470901489
x_t=28.9575177760	x_t=565730.6415795106
模式偏离函数的最终极小值	模式偏离函数的最终极小值
D*=0.9743678776	D*=0.9743678465

[a] 计算 D_0 所用的位置权重函数展开式系数的初始值是 a_1=1，a_2=2，a_3=3，a_4=4。

表 3-3 描述 *Escherichia coli* K12 基因组的学习样例，列出了：位置权重函数展开式系数的初始值、总模式偏离函数的初始值、极佳对偶描述子的参数值、最终获得的 D 值以及学习过程中交替的次数 [a]

学习样例 3	学习样例 4
位权函数展式系数初始值	位权函数展式系数初始值
a_1= 4.0000000000	a_1=-7886.1000000000
a_2= 2.0000000000	a_2=-12.0000000000

a_3= 1.0000000000	a_3=-0.8741009600
a_4= 3.0000000000	a_4=-487.0080000000
总模式偏离函数的初始值	总模式偏离函数的初始值
D_0=173.5196411111	D_0=469270199.4490945300
经过 2 次学习后，得到的极佳描述子	经过 5 次学习后，得到的极佳描述子
x_a= 0.0020963632	x_a=-0.0000001021
x_c= -0.0006435164	x_c=0.0000000313
x_g= 0.0103416484	x_g=-0.0000005038
x_t= -0.0178934133	x_t=0.0000008717
a_1= 0.2866388896	a_1=-5883.8711956368
a_2= 23.0962282583	a_2=-474108.1512971266
a_3= -0.0866386753	a_3=1778.1982796234
a_4= -0.2331481639	a_4=4785.7009585814
模式偏离函数的最终极小值	模式偏离函数的最终极小值
D*=0.9743678485	D*=0.9743678776

[a] 计算 D_0 所用的组成权重因子的初始值是 x_a =1，x_c =2，x_g =3，x_t =4。

表 3-4 使用公式（3-3）中的余弦基，描述 *Escherichia coli* K12 基因组的学习样例，列出了：组成权重因子和位置权重函数展开式系数的初始值、总模式偏离函数的初始值、极佳对偶描述子的参数值、最终获得的 D 值以及学习过程中交替的次数 [a]

学习样例 5	学习样例 6
组成权重因子初始值	位权函数展式系数初始值
x_a=1.0000000000	a_1=-7886.1000000000
x_c=2.0000000000	a_2=-12.0000000000
x_g=3.0000000000	a_3=-0.8741009600
x_t=4.0000000000	a_4=-487.0080000000
总模式偏离函数的初始值	总模式偏离函数的初始值
D_0=112.4425262112	D_0=234774673.1356592200
经过 3 次学习后，得到的极佳描述子	经过 14 次学习后，得到的极佳描述子
a_1=0.0003878983	x_a=-0.0000011425
a_2=0.0004087772	x_c=0.0000000707
a_3=0.0004480815	x_g=0.0000007493
a_4=0.0003707921	x_t=-0.0000005675
x_a=16.4466654669	a_1=-6075.2817969433
x_c=-1.1929718092	a_2=-5954.3111290240
x_g=-10.8348860880	a_3=-6302.9115983931
x_t=8.5727548802	a_4=-5139.2239660313
模式偏离函数的最终极小值	模式偏离函数的最终极小值
D*=0.9999278584	D*=0.9999279784

[a] 在学习样例 5 中，计算 D_0 所用的位置权重函数展开式系数的初始值是 a_1=1，a_2=2，a_3=3，a_4=4；在学习样例 6 中，计算 D_0 所用的组成权重因子的初始值是 x_a =1，x_c =2，x_g =3，x_t =4。

表 3-5 使用公式（3-2）中的余弦基，描述 *Caulobacter crescentus* 基因组的学习样例，列出了：组成权重因子和位置权重函数展开式系数的初始值、总模式偏离函数的初始值、极佳对偶描述子的参数值、最终获得的 *D* 值以及学习过程中交替的次数 [a]

学习样例 7	学习样例 8
组成权重因子初始值	位权函数展式系数初始值
x_a=1.0000000000	a_1=4.0000000000
x_c=2.0000000000	a_2=2.0000000000
x_g=3.0000000000	a_3=1.0000000000
x_t=4.0000000000	a_4=3.0000000000
总模式偏离函数的初始值	总模式偏离函数的初始值
D_0=111.7079456226	D_0=165.2392079028
经过 2 次学习后，得到的极佳描述子	经过 2 次学习后，得到的极佳描述子
a_1=-0.0000618397	x_a=0.0088980728
a_2=-0.0049388393	x_c=-0.0112783663
a_3=-0.0000097952	x_g=0.0098477830
a_4=0.0000280820	x_t=-0.0129567333
x_a=-39.4150073764	a_1=0.2736964748
x_c=49.9618155233	a_2=21.8781205256
x_g=-43.6240575385	a_3=0.0432996411
x_t=57.3953199658	a_4=-0.1242189356
模式偏离函数的最终极小值	模式偏离函数的最终极小值
D^*=0.9753179807	D^*=0.9753179899

[a] 在学习样例 7 中，计算 D_0 所用的位置权重函数展开式系数的初始值是 a_1=1，a_2=2，a_3=3，a_4=4；在学习样例 8 中，计算 D_0 所用的组成权重因子的初始值是 x_a =1，x_c =2，x_g =3，x_t =4。

讨论 表 3-2、表 3-3、表 3-4 和表 3-5 中共列出了 8 个学习样例。这里给出的只是对偶描述子的初始参数值和最终获得的极佳描述子的参数值及相应的模式偏离函数值。至于学习过程中的各参数的值及其变化，由于篇幅所限，没有列出。这些学习样例充分展示了对偶描述子交替式学习过程的特点：

其一，**唯一性**。模式偏离函数的最终极小值是唯一的，它与对偶描述子参数的初始值无关，也和进入交替学习过程的次序无关。表 3-2 和 3-3 中的 4 个学习样例，其参数初始值可谓千差万别，然而，最终所得到的 *D**却是一样的。这 4 个样例中，模式偏离函数的初始值 D_0 彼此相差很大，而最终所得到的 *D**值却直到小数点后第 8 位才有差异。我们可以认为这些 *D**值是一样的，而那微小的差距是由于计算过程中的舍入误差造成的。

其二，**不唯一性**。最终所得到的极佳描述子是不唯一的，它和对偶描述子参数的初始值有关，也和进入交替学习过程的次序有关。从这 4 个学习样例中可以看出，最终所获得的极佳对偶描述子彼此很不相同，它是由对偶描述子的初始值和进入交替式学习过程的次序决定的。另外，学习过程的长短，即收敛到 *D**值所经历的学习步骤和程序的运行时间，也和对偶描述子的初始参数值有关。

对偶描述子学习的唯一性，是由模式偏离函数的定义决定的。因为模式偏离函数是以欧氏距离的方式定义的二次函数，它在定义域上只有唯一的全局极小，所以，最终获得的模式偏离函数极小值是唯一的。**对偶描述子学习的不唯一性，是由模式描述函数的定义决定的**。因为模式描述函数定义成组成权重因子和位置权重函数两项的乘积，而由积不能决定两个因子，因此，最终获得的极佳对偶描述子是不唯一的。对偶描述子学习的这种**唯一性又不唯一性**充分体现了对立统一的辩证规律。

最终获得的模式偏离函数极小值是唯一的，它究竟是多少，取决于两个方面：（1）序列本身的特征，这里即为原核基因组蛋白质编码区的公共特征的多少和其提取的难易程度；（2）所选择的基函数的逼近能力，这里就是不同的三角余弦周期基的逼近能力。表 3-2 和表 3-3 中的 *D**之所以是相同的，在于它们选择了相同的字符序列 （都是 *E. coli* K12 的基因组），也选择了相同的逼近基函数（3-2）。

表 3-4 中，物种仍然是大肠杆菌 K12 株，其最终 *D**值之所以不同于表 3-2 和表 3-3 的，是因为表 3-4 中选择的是另一组基函数（3-3）。表 3-4 中的 *D**值大于表 3-2 和表 3-3 的中的 *D**值，说明，在这个问题中，（3-3）式的逼近能力不如（3-2）式的逼近能力，进一步说明了在 *E. coli* K12 的基因组中 2、3、4、5 周期性的联合使用要比 4、5、6、7 周期性的联合使用更能反映蛋白质编码区的特征。

表 3-5 中，基函数仍然是（3-2），其最终 *D**值之所以不同于表 3-2 和表 3-3 的，是因为表 3-5 中选择的是另一个物种的基因组（*Caulobacter crescentus*）。表 3-5 中的 *D**值大于表 3-2 和表 3-3 的中的 *D**值，说明，用（3-2）式给出的基函数对物种 *Caulobacter crescentus* 基因组的特征的逼近效果不如对物种 *Escherichia coli* K12 的特征的逼近效果，进一步说明了物种 *Escherichia coli* K12 中的 2、3、4、5 周期性要强于物种 *Caulobacter crescentus* 中的 2、3、4、5 周期性。

3.1.3 二阶对偶描述子的交替式学习过程演示

作为高阶对偶描述子的例子，这里看一下二阶对偶描述子。在第二章中对偶描述子应用

扩展部分提到，高阶对偶描述子的组成权重因子考虑了多字符的关联。二阶对偶描述子考虑的是双字符的关联。对于 DNA 序列来说，由于有 4 种碱基的缘故，其二阶对偶描述子就有 4^2=16 个组成权重因子。

材料与方法 材料仍然用大肠杆菌 K 12 株的基因组，逼近位权函数仍然用（3-2）式。交替式学习过程中，所用的公式与上面一阶描述子情形下的相同。这里，不同于前面的地方在于，高阶对偶描述子存在两种描述方式：线性描述和非线性描述。对于二阶对偶描述子来说，线性描述每次从左向右滑动 1 个字符，而非线性描述方式则每次滑动两个字符。关于这两种描述方式的详细描述，参看第二章中的相关内容。对于前面的一阶描述子来说，线性描述和非线性描述这两种描述方式是重合的，即是一回事儿，不必区分。

结果 在表 3-6 和表 3-7 中给出的学习样例 9-12，就是二阶对偶描述子的学习过程的初始值和最终结果，中间过程的详细信息没有列出。表 3-6 中的学习样例使用的是线性描述，表 3-7 中的则是非线性描述方式。

表 3-6 二阶对偶描述子的学习样例，使用线性描述方式

学习样例 9	学习样例 10
对偶描述子的初始值	对偶描述子的初始值
x_{AA}=1.0000000000	a_1=0.8414709848
x_{AC}=2.0000000000	a_2=90.2365955020
x_{AG}=3.0000000000	a_3=-647.0672563708
x_{AT}=4.0000000000	a_4=-1585.2483991630
x_{CA}=5.0000000000	x_{AA}=1.0000000000
x_{CC}=6.0000000000	x_{AC}=2.0000000000
x_{CG}=7.0000000000	x_{AG}=3.0000000000
x_{CT}=8.0000000000	x_{AT}=4.0000000000
x_{GA}=9.0000000000	x_{CA}=5.0000000000
x_{GC}=10.0000000000	x_{CC}=6.0000000000
x_{GG}=11.0000000000	x_{CG}=7.0000000000
x_{GT}=12.0000000000	x_{CT}=8.0000000000
x_{TA}=13.0000000000	x_{GA}=9.0000000000
x_{TC}=14.0000000000	x_{GC}=10.0000000000
x_{TG}=15.0000000000	x_{GG}=11.0000000000
x_{TT}=16.0000000000	x_{GT}=12.0000000000
a_1=1.0000000000	x_{TA}=13.0000000000
a_2=2.0000000000	x_{TC}=14.0000000000
a_3=3.0000000000	x_{TG}=15.0000000000
a_4=4.0000000000	x_{TT}=16.0000000000

总模式偏离函数的初始值	总模式偏离函数的初始值
D_0=1459.0168336139	D_0=139060875.6466771700
经过 2 次学习后，得到的极佳描述子	经过 2 次学习后，得到的极佳描述子
a_1=-0.0000303778	x_{AA}=0.0000044196
a_2=-0.0047211773	x_{AC}=-0.0000001936
a_3=-0.0000077480	x_{AG}=-0.0000224100
a_4=0.0000147008	x_{AT}=0.0000136128
x_{AA}=-20.8926758021	x_{CA}=0.0000047591
x_{AC}=0.9140403168	x_{CC}=-0.0000077024
x_{AG}=105.9184807144	x_{CG}=-0.0000180360
x_{AT}=-64.3857999497	x_{CT}=0.0000162189
x_{CA}=-22.4939723036	x_{GA}=0.0000236607
x_{CC}=36.3837148465	x_{GC}=0.0000042000
x_{CG}=85.2473266022	x_{GG}=0.0000042491
x_{CT}=-76.6590941875	x_{GT}=0.0000096951
x_{GA}=-111.6729341485	x_{TA}=-0.0000144598
x_{GC}=-19.8249861170	x_{TC}=-0.0000200562
x_{GG}=-20.0412672524	x_{TG}=-0.0000438553
x_{GT}=-45.8063579457	x_{TT}=0.0000013074
x_{TA}=68.3316218881	a_1=143.5969571288
x_{TC}=94.7707149841	a_2=22309.1435246135
x_{TG}=207.2985156097	a_3=36.4256514572
x_{TT}=-6.1737033823	a_4=-69.6071118440
模式偏离函数的最终极小值	模式偏离函数的最终极小值
D^*= 0.9383110128	D^*=0.9383103521

表 3-7 二阶对偶描述子的学习样例，使用非线性描述方式

学习样例 11	学习样例 12
对偶描述子的初始值	对偶描述子的初始值
x_{AA}=1.0000000000	a_1=0.8414709848
x_{AC}=2.0000000000	a_2=90.2365955020
x_{AG}=3.0000000000	a_3=-647.0672563708
x_{AT}=4.0000000000	a_4=-1585.2483991630
x_{CA}=5.0000000000	x_{AA}=1.0000000000
x_{CC}=6.0000000000	x_{AC}=2.0000000000
x_{CG}=7.0000000000	x_{AG}=3.0000000000
x_{CT}=8.0000000000	x_{AT}=4.0000000000
x_{GA}=9.0000000000	x_{CA}=5.0000000000
x_{GC}=10.0000000000	x_{CC}=6.0000000000
x_{GG}=11.0000000000	x_{CG}=7.0000000000
x_{GT}=12.0000000000	x_{CT}=8.0000000000
x_{TA}=13.0000000000	x_{GA}=9.0000000000

x_{TC}=14.0000000000	x_{GC}=10.0000000000
x_{TG}=15.0000000000	x_{GG}=11.0000000000
x_{TT}=16.0000000000	x_{GT}=12.0000000000
a_1=1.0000000000	x_{TA}=13.0000000000
a_2=2.0000000000	x_{TC}=14.0000000000
a_3=3.0000000000	x_{TG}=15.0000000000
a_4=4.0000000000	x_{TT}=16.0000000000
总模式偏离函数的初始值	总模式偏离函数的初始值
D_0=1460.4148256887	D_0=139024249.8476721900
经过 4 次学习后，得到的极佳描述子	经过 2 次学习后，得到的极佳描述子
a_1=-0.0000084290	x_{AA}=0.0000048180
a_2=-0.0046561896	x_{AC}=-0.0000003169
a_3=-0.0000314787	x_{AG}=-0.0000244458
a_4=-0.0000537996	x_{AT}=0.0000154551
x_{AA}=-21.5198041020	x_{CA}=0.0000050887
x_{AC}=1.3985053222	x_{CC}=-0.0000084734
x_{AG}=109.3924566572	x_{CG}=-0.0000194130
x_{AT}=-69.2848463333	x_{CT}=0.0000172763
x_{CA}=-22.7493687198	x_{GA}=0.0000250877
x_{CC}=37.8868620404	x_{GC}=0.0000044609
x_{CG}=86.8869751832	x_{GG}=0.0000041842
x_{CT}=-77.1631515153	x_{GT}=0.0000103277
x_{GA}=-111.7886031821	x_{TA}=-0.0000151778
x_{GC}=-19.8754642879	x_{TC}=-0.0000218199
x_{GG}=-18.6585348639	x_{TG}=-0.0000461231
x_{GT}=-46.1804788806	x_{TT}=0.0000012138
x_{TA}=67.8272895506	a_1=37.2089026411
x_{TC}=97.5622254226	a_2=20816.0108801573
x_{TG}=206.3846637137	a_3=140.7176657131
x_{TT}=-5.4619034576	a_4=240.5242664844
模式偏离函数的最终极小值	模式偏离函数的最终极小值
D*=0.9410114750	D*=0.9410113138

讨论 从表 3-6 和表 3-7 中可以看出，前面讲到的关于对偶描述子学习的唯一性与不唯一性都得到了充分的体现。表 3-6 中列出的学例 9，是首先给定一组组成权重因子的值，然后进入交替式学习过程，而在表 3-7 中，学例 10 则是首先给定一个位置权重函数，而后进入学习过程。结果，它们表现得与一阶对偶描述子一样，收敛到相同的 D*值。这就是说，对偶描述子交替式学习的**唯一性又不唯一性**，对于各阶描述子都是成立的。（注意，这里所说的相同的 D*值，是指对于二阶对偶描述子不同的初始参数来说，最终所获得的 D*值是相

同的，而并不是说二阶对偶描述子所获得的最终 D^*值与一阶描述子最终获得的 D^*值相同。事实上，它们一般不相同。）

比较表 3-6 和表 3-7 中的结果，便会知道，对于高阶对偶描述子来说，最终取得的模式偏离函数值是依赖于所使用的描述方式的。表 3-6 中的线性描述方式所得的 D^*值，要小于表 3-7 中的非线性描述方式，说明线性描述方式对字符序列的刻画更加细致逼真。高阶对偶描述子又丰富了对偶描述子学习的不唯一性，即对于不同的描述方式，最终所取得的 D^*值不唯一。

3.2 对偶描述子用于 DNA 序列蛋白质编码区的识别

基因编码区特征的提取与基因识别是一个问题的两个方面。找到了基因编码区的特征，就可以依据待识别的序列是否具有该特征来判定它是不是编码区。DNA 序列蛋白质编码区的识别就是通常所说的狭义的基因识别，而广义的基因识别是指基因完整结构的识别。由于基因组测序工程的大量开展，国际公共核酸序列数据库中的 DNA 序列数据越来越多，仅靠实验的方法确定其中的基因及其位置，在时间和金钱两方面都是不堪承受的。于是，计算机辅助基因识别便成为了计算生物学（生物信息学）领域中的一个重要课题。计算机辅助基因识别 (gene identification, gene finding, or gene recognition) 的基本问题是在给定基因组序列后，正确识别基因的范围和其在基因组序列中的位置。经过二十余年的努力，现已提出了数十种预测蛋白质编码基因的算法，其中，有十种左右重要的算法及相应软件提供了免费的网上服务[135]。

计算机辅助基因识别算法大体上可分成两类：基于序列同源性的基因识别和基于序列统计特征的基因识别。基于序列同源性的基因识别，使用序列比对工具，如 BLAST 或 FASTA 等来搜索核酸序列无冗余数据库中的已知序列，并依据同源性大小（相似程度），来确定待识别序列的基因位置与功能。例如，识别工具 ORPHEUS[136] 主要基于序列同源性，同时考虑了密码子使用，识别工具 CRITICA[137] 主要基于序列比较信息，再辅之以六核苷酸使用。

基于序列统计特征的基因识别算法，主要利用蛋白质编码区的组成特性和一些功能位点的保守信号。早期的工作对 DNA 序列组分[138]、密码子使用[139] 和氨基酸使用[140]进行统计分析，并将所得的统计特征量用于识别蛋白质编码区。后续研究表明，蛋白质编码区所特有的许多性质如 3 周期性[141]、密码子位 G＋C 含量偏向性[142]、密码子使用的偏向性等都可用于蛋白质编码区的识别。文献[143]列出了当时关于基因识别的二十来种算法，并

认为六核苷酸使用 (hexamer usage) 是描述蛋白质编码区的最好方法。

近年来，基因识别算法有了较大的发展，开发出来许多实用的程序。对于细菌和古细菌等原核基因组，现有的著名的基因识别算法和程序有 GeneMark 系列[31,32,36]、Glimmer 系列[33-35]和 GeneHacker Plus[144]。其中，目前使用最为广泛的程序是 Glimmer。这些算法是以高阶马尔科夫模型 (higher-order Markov chain model) 或隐马尔可夫模型 (hidden Markov model) 为其理论基础的。最近又推出了用于原核和冠状病毒基因识别的 Zcurve 系列软件[52,53]。Zcurve 系列是以我国学者张春霆等人提出的 DNA 序列的等价的几何表示理论——Zcurve 理论为基础开发的。该理论用于基因识别时，很强地依赖于先验知识，即相位特异性（实际上就是编码区的 3 周期性），但以其简单直观、参数少（因而所需的训练样本集相对较小）等优点，大有后来居上之势。现有的真核生物基因识别程序有 GeneID[145]、MZEF[70]、Genscan[146]、GeneMark.hmm[32]、GlimmerM[35]、AUGUSTUS[147]等。

当前，原核生物基因识别中存在的一些难题主要有：(i) 原核生物基因没有内含子，但是基因间隔很少，基因容易发生重叠 (比如，重叠 4bp 或 13bp)，基因 5’ 端的翻译起始位点很难正确预测。(ii) 短基因 (比如，长度 $< 60 \sim 80$ 氨基酸残基) 的组成特征不明显，统计模型难于正确识别。(iii) 统计模型过度依赖于训练集，对碱基组成“非典型”的基因 (比如，水平转移基因 (horizontally transferred genes)，识别率低。(iv) 对于一些基因组，尤其是高 G+C 含量的基因组，伪正率过高。对于真核生物基因识别来说，由于内含子的存在，其基因结构要比原核生物的复杂，因此，正确识别出起始密码子、剪切位点、终止密码子和完整的基因结构，是相当困难的。现有的真核生物基因识别算法在核苷酸水平识别率较高，达 90%；但是，在外显子水平识别的正确率较低，小于 50%[148]。因此，其研究现状不能令人满意，仍有大量的工作要做[135]。

基于序列同源性的基因识别和基于序列组成特征的基因识别各有优缺点。前者识别率较低，例如，对于一个新测序的细菌基因组，只有约 60~70%的基因在当前数据库中有同源序列，约 30~40%则是新基因，不能在已知的基因数据库中找到。后者的识别率相对较高，但存在过度依赖于训练集、伪正率高等问题。如能将两类方法结合使用，互补余缺，基因识别成绩可能更好。例如，基因识别工具 BDGF[149]结合序列同源性和统计学特征，取得了较好的预测成绩。

关于基因识别方面的综述文献较早可以参看[143,150-152]，较近的则有[148,153-155,135]。

3.2.1 对偶描述子用于原核基因识别

材料与方法

材料 这里所用的基因组数据包括了前面的大肠杆菌的 K12 株和物种 *Caulobacter crescentus*，此外，还包括了低等真核生物酿酒酵母 *Saccharomyces cerevisiae*，俗称 Yeast。Yeast 的基因注释信息是 2003 年 3 月，从 http://speedy.mips.biochem.mpg.de 下载的。这里只用到了它的第一类 ORF，包括 3275 个已知功能的基因。酿酒酵母 Yesat 作为一个低等真核生物，它的基因结构较为简单，非常接近原核生物。也就是说，它的基因组中内含子很少，因此，具有较完整的 ORF。

方法 对偶描述子方法本身是比较完善的，它可以直接用于序列的识别，而无需借助其它方法，但是，在有效地结合了其它的一些识别算法，如 Fisher 判别等方法后，可能会相得益彰，取得更好的效果。为了检验算法，需要正负两类样本。正样本就由 Genbank 数据库中的注释基因充当，而负样本的产生则采用了随机打乱的方法。具体做法是把相应的正样本随机打乱一万次，这样所得到的负样本的数量与正样本的数量是相同的，并且在碱基组成上负样本也和与它相应的正样本一样，只是排列方式不同。

判别方法 1：**d 值阈值判别法**。这是对偶描述子本身固有的方法。极佳对偶描述子对应于模式偏离函数的极小值，相当于专门针对原序列（训练数据）做成的一件衣服，让待识别序列来试穿这件衣服，就是用极佳对偶描述子对待识别序列从头到尾描述一遍，以计算它的模式偏离函数的取值 d。对 d 设定一个阈值（通常依据伪正等于伪负），则待识别序列是否和原序列同属一类，就是看其 d 值是否小于该阈值。首先，针对各物种的注释基因（正样本）使用交替式学习方式来训练对偶描述子。然后将这些正负样本掺杂在一起，用极佳对偶描述子去计算它们的模式偏离函数值。接下来，利用伪正率等于伪负率来设置 d 的阈值，并根据该阈值算出敏感度、特异度和准确率。

判别方法 2：**加权频率 Fisher 判别**。对偶描述子实现了加权统计的思想，可以通过位置加权频率来反映字符序列的排列信息，因此，可以区分不同类型的字符序列。只要对位置适当加权后，就可以利用统计所得的加权频率来表征字符序列。这里并不限定位权函数的获得方式。位权函数可以利用先验信息而预先给定，也可以根据一定的样本数据训练得到。在给定了一个位置权重函数后，就可以应用公式(2-7)提取对偶变量的排列部，对长度归一化后，得到加权频率作为字符序列的特征量。有了特征量，就可以运用 Fisher 判别方法，构造 Fisher 线性判别函数和判别准则，进行识别了。关于 Fisher 判别的内容，可以参看文献[69]和[51]。

这里值得一提的是，由于提取对偶变量的排列部时，假定了组成平权，同时，这里又给定了位置权重函数，因此，各字符的加权频率就不独立了。它们之间存在一个约束条件，即各字符的加权频率之和等于位置权重函数在整个序列长度区间上的积分值。因此，这里在 A、C、G、T 四个字符的加权频率中任取 3 个即可。

识别结果的好坏，用敏感度（sensitivity）、特异度（specificity）和准确率(accuracy)来衡量。其中，敏感度定义成正确识别出来的正样本的数目与正样本总数目的比值，特异度定义成正确识别出来的负样本的数目与负样本总数的比值，而准确率定义成它俩的平均值，即，

$$\text{Accuracy}=(\text{Sensitivity}+\text{Specificity})/2 \tag{3-4}$$

上式中的 sensitivity 和 specificity 又简称 Sn 和 Sp。关于它们的详细讨论，参见第五章。

结果与讨论 将对偶描述子用于原核生物基因识别的结果列于表 3-8 和 3-9 中。先看表 3-8。表 3-8 是使用对偶描述子固有的判别方法：d 值阈值判别法，所给出的结果。下面逐行对其进行说明。第 1 行是用 *E. coli* K12 的注释基因作为训练数据，采用交替式学习，训练得到的一阶极佳对偶描述子（学习样例 1 中给出）进行识别的结果。第 2 行是以物种 *Caulobacter crescentus* 的基因组和相应的注释信息来训练的对偶描述子（由学习样例 9 给出），所给出的识别结果。前面提到，对物种 *Caulobacter crescentus* 基因组的特征的逼近效果不如对物种 *Escherichia coli* K12 特征的逼近效果，同样，在这里又看到，用学习样例 9 中所得到的极佳对偶描述子进行识别的结果就比用学习样例 1 中的差。第 3 行是低等真核生物 Yeast 的识别结果，这里选用它的第一类 ORF 作为训练数据，然后，用 d 值阈值判别法进行识别。第 4 行是使用 *E. coli* K12 的极佳对偶描述子，对物种 *Caulobacter crescentus* 的蛋白质编码区进行跨物种的基因识别，结果准确率也达到 90%以上。第 5 行，同样是用 *E. coli* K12 的极佳对偶描述子，但是对真核生物 Yeast 进行跨界基因识别，结果准确率也在 90%以上，尚且略高于第 4 行中的结果。这说明，自然界中各物种的蛋白质编码区之间存在着很强的共性。这是因为，它们都使用同一套密码子的缘故。

第 6、7 两行，是使用二阶描述子的 d 值阈值判别法对 *E. coli* K12 进行识别的结果。其中，第 6 行用的是线性描述方式，第 7 行用的是非线性描述方式。第 7 行的准确率比第 1 行的准确率略高，说明用高阶对偶描述子对字符序列进行描述，由于组成权重因子个数多的缘故，其精细程度要高于一阶的情形。第 6 行的准确率是最高的，这可能是由于它采用的高阶对偶描述子的线性描述方式，在一定的程度上，综合高阶描述子的参数多和线性描述刻画精细这两者的优点的缘故。

表 3-8 使用 d 值阈值判别法对各物种的蛋白质编码区、非编码区进行识别，所得的敏感度、特异度和准确率及相应的阈值 [a]

Species	ORF number	Threshold	Sensitivity (%)	Specificity (%)	Accuracy (%)
Ecoli K12	4289	1.005950	95.06	94.96	95.01
C. cre	3737	1.008200	93.90	94.14	94.02
Yeast	3275	1.007800	94.08	93.71	93.90
C. cre^{E}	3737	1.007410	91.81	91.89	91.85
Yeast^{E}	3275	1.013600	93.38	92.43	92.90
Ecoli K12^{o2l}	4289	1.014760	97.06	97.34	97.20
Ecoli K12^{o2n}	4289	1.016280	95.36	95.64	95.50

[a] C. cre^{E} 和 Yeast^{E} 分别表示使用物种 *E. coli* K12 的极佳对偶描述子来识别物种 *Caulobacter crescentus* 和物种 Yeast 的蛋白质编码基因；^{o2l} 和 ^{o2n} 分别表示使用 2 阶对偶描述子的线性描述方式和非线性描述方式来识别蛋白质编码基因。

为了对识别结果有一个直观的印象，可以看一下图 3.1（对应于表 3-8 的第 1 行）。

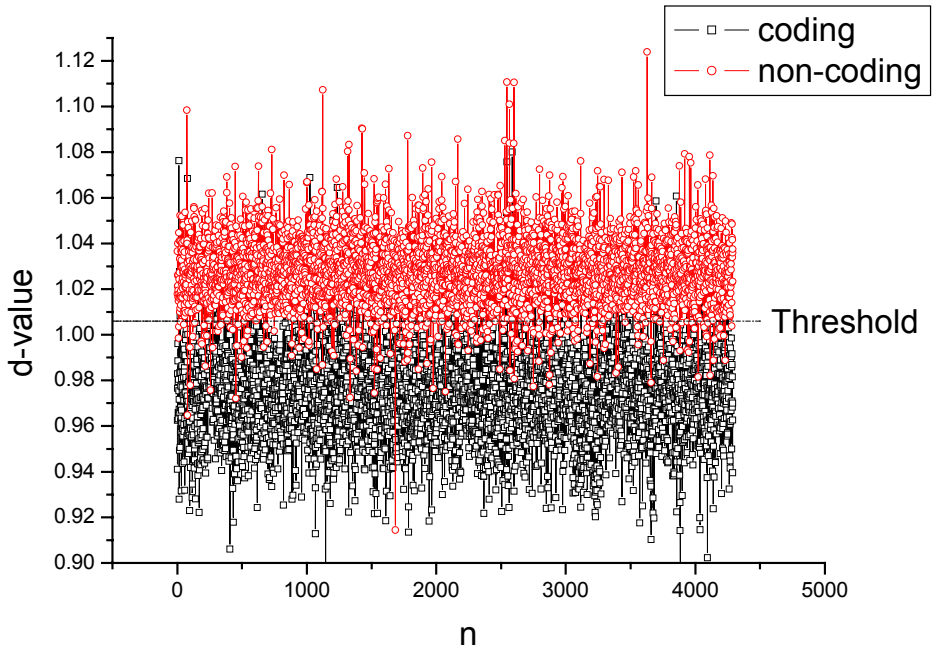


图 3.1 用 d 值阈值判别法，对大肠杆菌 *E. coli* K12 的基因识别效果

图中，横坐标表示基因数目，纵坐标表示每条序列的模式偏离函数值；图中的虚线表示设定的阈值；虚线以上的红色圆圈表示非编码序列，虚线以下的黑色方块表示编码序列。

再看表 3-9。表 3-9 用的是加权频率 Fisher 判别法。表中列出的是对大肠杆菌基因组编

码区与非编码区进行识别，十次交叉检验的平均的敏感度、特异度和准确率。所谓交叉检验，就是拿训练集中的一部分数据来训练，拿另一部分来检验。具体到这里，就是拿 *E. coli* K12 的注释基因的一半作为训练集，来获取 Fisher 系数，并用剩下的一半，作为检验集来检验识别结果。第 1 行是依据先验知识：蛋白质编码区存在 3 周期性而人为构造的一个函数。该函数通过整数位置 k 对 3 取模的方式把周期性为 3 的一个干扰引入字符序列的统计过程。由于蛋白质编码区存在 3 周期性，所以能够与这个外力发生共振，从而使某些字符的加权频率得到增强，而另一些字符的加权频率减弱。而非编码的随机序列不存在明显的 3 周期性，因此，不能够与它发生共振，故而，各字符的加权频率彼此相差不大。这样一来，用加权频率作为特征量就可以把两类字符区分开来。第 2-4 行分别用到了学习样例 1、9 和 11 中的极佳对偶描述子的位置权重函数。第 2 行中用学习样例 1 中一阶极佳描述子的位置权重函数对字符序列中的各位置赋权，最终得到的加权频率是 4 个实数，由于它们彼此不独立，存在一个约束条件，因此，只需取其中的 3 个来表征字符序列。数字实验表明，对于不同的取法，所得 Sn 和 Sp 会微有差异，但是平均后所得的准确率是一样的。这在某种意义上提供了一种灵活性，使我们可以在高识别率与低伪正率之间，根据实际需要，权衡得失，做出选择。

第 3 行和第 4 行由于使用了二阶描述子，其独立参数的个数都是 15 个。第 3 行的线性描述方式，由于刻画细致，所得的识别率较第 4 行的非线性描述方式略高。而第 3 行和第 4 行的准确率又都高于第 2 行，说明，高阶描述子在使用加权频率 Fisher 判别时，由于参数相对较多的缘故，要优于低阶描述子所给出的结果。但是，这种“高”也不是无条件的，因为，多参数意味着需要更多的训练样本，因此，只有在样本充足的情况下，才允许使用高阶的对偶描述子。这也提供了一种灵活性，即可以依据样本量的大小来选择适当阶数的对偶描述子。

表 3-9 加权频率 Fisher 判别对大肠杆菌基因组编码区与非编码区的识别结果，十次交叉检验的平均的敏感度、特异度和准确率

Species	ORF number	$I(k)$	Sensitivity (%)	Specificity (%)	Accuracy (%)
Ecoli K12[ap]	4289	$(-e)^{k \bmod 3}$	93.84±0.23	91.71±0.33	92.78
Ecoli K12	4289	学习样例 1	92.92±0.29	96.70±0.23	94.80
Ecoli K12	4289	学习样例 9	95.72±0.13	99.23+0.11	97.48
Ecoli K12	4289	学习样例 11	93.96±0.26	97.72±0.15	95.84

[ap] a priori 表示依据先验知识给定位置权重函数：$(-e)^{k \bmod 3}$。

为了对识别结果有一个直观的印象，可以看一下图 3.2（对应于表 3-9 的第 3 行）。

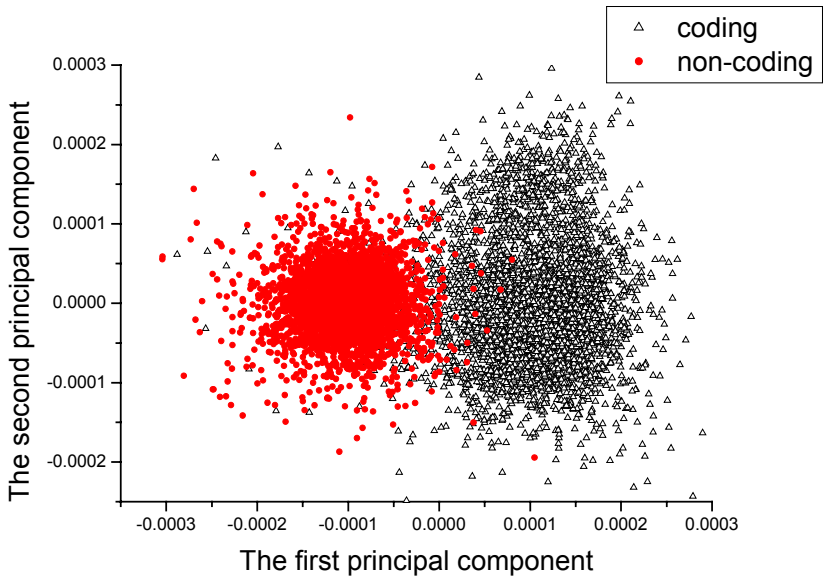


图 3.2 使用加权频率 Fisher 判别法，对大肠杆菌 *E. coli* K12 的基因识别效果

这里对前 15 个加权频率进行了主成分分析。图中，横坐标表示第 1 主成分，纵坐标表示第 2 主成分，两者加起来占 51.13%；图中红色实心圆圈表示非编码序列，黑色空心三角表示编码序列。

3.2.2 在人类基因组外显子和内含子识别中的应用

材料与方法

材料 这里所用到的人类基因组外显子数据摘自下面的文件 ftp://genome.lbl.gov/pub/genesets/HUMAN/4813_Hum_CDS.fa，其发布时间为 1999 年 6 月 5 日。这个文件里包含了 4813 个带有起始密码子和终止密码子的完整人类基因编码序列。内含子数据则摘自同一 FTP 网址 intron_v105 目录下的文件，其中包括了 462 个人类基因。从上面的这些文件中，共抽取了 4000 个长度为 210 的外显子片断和同等数量同样长度的内含子片断。任何两个相邻的序列之间彼此没有重复。外显子片断作为正样本，内含子片断作为负样本，构成了两个数据集合。

方法 这里除了用到前面介绍的 d 值阈值判别和加权频率 Fisher 判别外，还用到了两类 d 值判别。关于 d 值阈值判别和加权频率 Fisher 判别前面已经讲过，这里只说一说两类 d 值判别。这里的两类 d 值判别是作为对偶描述子多类识别的一个例子给出的。对于两个数据集中的正负两类样本，分别应用交替式学习，训练对偶描述子的参数。用所得到的极佳对偶描述子，分别去描述待识别的序列，从而得到两个 d 值：d_1 和 d_2，分别对应属于两类的“概率”，然后应用公式（2-34）依据 d_1 和 d_2 的大小来确定待识别序列所属的类别。即：若 $d_1<d_2$，

则待识别序列属于第一类即外显子；若 $d_2<d_1$，则待识别序列属于第二类即内含子。

结果与讨论

结果 表 3-10 到表 3-12 给出了用对偶描述子进行人类基因组外显子和内含子识别的例子。其中，表 3-10 用的是依据蛋白质编码区具有 3 周期性的先验知识给出的位置权重函数，共有 3 个，统一取作 $(-e)^{(k+a) \bmod 3}$ 的形式。表 3-10 的第 1-3 行中的位置权重函数其参数 a 的取值分别为-1、0、+1，相应于整数除 3 的三个可能的余数。表 3-10 的第 4 行，是同时使用这三个位置权重函数时，用加权频率 Fisher 判别所给出的结果。

表 3-10 先验给定位置权重函数，使用加权频率 Fisher 判别对人类基因组的外显子和内含子进行识别的十次交叉检验的平均敏感度、特异度和准确率

$I(k)$	Sensitivity (%)	Specificity (%)	Accuracy (%)
$(-e)^{(k-1) \bmod 3}$	71.34±0.49	85.22±0.44	78.28
$(-e)^{k \bmod 3}$	62.68±0.32	65.30±0.76	63.99
$(-e)^{(k+1) \bmod 3}$	78.30±0.49	73.64±0.55	75.97
All of the above	87.58±0.28	91.34±0.70	89.46

表 3-11 是使用不同类型的对偶描述子或描述方式或判别方法而得到的结果。下面分别说明。第一版块，即前三行，使用的是一阶描述子。从准确率上看，在 d 值阈值判别、两类 d 值判别和加权频率 Fisher 判别等三种判别方法中，d 值阈值判别最高，两类 d 值判别次之，加权频率 Fisher 判别更次之，且三种判别方法之间彼此相差不大，均徘徊于 75%左右。第 2 版块，即中间三行，使用的是二阶描述子的线性描述方式。在这一版块中，准确率的次序与第 1 版块恰好相反：加权频率 Fisher 判别的效果最好，两类 d 值判别次之，而 d 值阈值判别尤次之，但三种判别方法之间彼此相差不大，均徘徊于 87%左右，却明显高于第 1 版块中的结果。第 3 版块中使用的是二阶对偶描述子的非线性描述方式，其结果总体上介于第 1 版块和第 3 版块之间，平均在 83.5%左右。具体来说又以加权频率 Fisher 判别为最高，而 d 值阈值判别和两类 d 值判别的效果则不相上下。

表 3-11 依据数据训练得到的各阶对偶描述子，使用多种判别方法对人类基因组外显子区和内含子区的识别结果

	判别方法	Sensitivity (%)	Specificity (%)	Accuracy (%)
一阶描述子	*D* 值阈值判别	75.80	76.38	76.09
	两类 *D* 值判别	72.55	78.95	75.75
	加权频率 Fisher 判别	72.67	76.73	74.70
二阶线性	*D* 值阈值判别	86.70	87.00	86.85
	两类 *D* 值判别	84.70	90.02	87.36
	加权频率 Fisher 判别	85.06	92.14	88.60
二阶非线性	*D* 值阈值判别	83.40	83.52	83.46
	两类 *D* 值判别	81.78	84.88	83.33
	加权频率 Fisher 判别	82.34	86.59	84.46

表 3-12 同时使用了表 3-10 中的 3 个先验的位置权重函数和表 3-11 中的二阶对偶描述子的两种描述方式，给出了对于不同长度的人类基因组外显子与内含子片断，应用加权频率 Fisher 判别的结果。作为**多对偶描述子**的例子，这里相当于使用了 5 个对偶描述子：3 个一阶的和 2 个二阶的，总共是 39 个独立参数，其中，3 个一阶，每个 3 个，共 9 个，2 个二阶，每个 15 个，共 30 个。可以看出，随着序列长度的变短，识别的准确率是逐渐降低的，这是由于短序列统计特征不明显的缘故。但另一方面，也可以看出这种下降并不快，这是由于在判别时已经对序列的长度进行了归一化的缘故。

表 3-12 结合先验知识与后验训练数据，对人类基因组不同长度的外显子和内含子的片段进行识别，十次交叉检验的平均的敏感度、特异度与准确率

Fragment length(bp)	192	162	129	108	87	63	42
Sensitivity	89.56	88.66	87.71	86.64	84.82	82.86	81.36
Specificity	91.64	90.83	87.88	86.88	86.10	82.35	79.75
Accuracy	90.60	89.74	87.80	86.76	85.46	82.60	80.56

讨论 从表 3-11 和表 3-12 还可以看出，对偶描述子不仅存在多种判别方法，而且，还能够把先验知识和后验数据结合起来使用。多种判别方法的存在，为特定问题的解决，提供

了选择的余地。在对偶描述子及其描述方式确定了的情况下，使用不同的判别方法，所得的识别结果会略有差异，但相去无几。因此，关键就不在判别方法了，而在于对偶描述子和它的描述方式的选择。如表 3-11 所示，高阶对偶描述子的线性描述方式，由于综合了参数多和位置刻画细致等优点，所得到的识别率最高。在训练样本集足够大的情况下，高阶描述子的识别结果要优于低阶描述子的识别结果。如果，训练样本集不够大，则有可能在对偶描述子学习过程中出现矩阵奇异，线性方程组无法求解的情况。另外，高阶对偶描述子的线性描述方式，在样本的“典型性”不够（即在样本空间中的分布，与总体的样本分布相比，较偏），则有可能出现过度训练的情况，从而影响对偶描述子预测的泛化能力。

作为对偶描述子与其他判别方法结合使用的例子，这里介绍了加权频率 Fisher 判别。加权频率，由于位置权重函数的引入，可以优于常规的频率而反映组成和排列两方面的信息，因此，可以作为字符序列的特征量使用。至于，位置权重函数的获取方式，方法本身对此并无限制。如果，有**先验信息**可资利用，像本例中的蛋白质编码区的 3 周期性，则可以依据先验信息，构造位置权重函数。相反，若无先验信息，而只有一些采样得到的**后验数据**，也没关系，对偶描述子方法的学习机制可以实现从 Raw Data 中的知识挖掘。

从表 3-10 可以看出，多对偶描述子联合使用的威力。前 3 行里都是单独使用一个位置权重函数的识别结果，它们或高或低，但总体说来都不太高。然而，将这三个位置权重函数联合起来使用，每一对偶描述子描述序列的一部分特征，而它们的全体可以给出一个相对较为完整的描述，因此，效果就会显著变好。表 3-12 则表明，先验知识和后验学习可以联合使用以进一步提高识别率。表 3-12 中的例子，共使用了 3 个先验给定的位权函数和两个学习得到的位权函数，加起来，相当于共使用了 5 个对偶描述子。可以看出，它们联合使用的效果要好过每一个单独使用的效果。

总之，对偶描述子以加权统计的思想和较为完善的学习机制，使得先验知识的利用和基于后验数据的学习联合起来成为可能，再辅之以多种判别方法的运用，可以相当灵活地表现相当强大的功能。

参考文献

1. 郭锋彪，原核生物基因识别程序 ZCURVE 1.0 的发展及基因组序列分析：[硕士学位论文]，天津；天津大学，2002
2. International Human Genome Sequencing Consortium. (2001) Initial sequencing and analysis of the human genome. Nature, **409**, 860-921
3. Fleischmann, R.D., Adams, M.D. and White O. *et al*. (1995) Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. Science, **269**, 496-512
4. Kyrpides N., Genomes OnLine Database (GOLD 1.0) (1999) a monitor of complete and ongoing genome projects world-wide, Bioinformatics, **15**, 773-774
5. David W. Mount, Bioinformatics: Sequence and Genome Analysis, 北京：科学出版社（英文影印版），2002
6. Minoru Kanehisa(金久时)　Post-genome informatics(后基因组信息学)（孙之荣等译），北京：清华大学出版社，2002
7. 郝柏林、张淑誉，生物信息学手册（第二版），上海：上海科学技术出版社，2002
8. 来鲁华等，蛋白质的结构预测与分子设计，北京：北京大学出版社，1993
9. Kabsch, W. and C. Sander (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers **22**: 2577-2637.
10. Richards, F. M. and C. E. Kundrot (1988). Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. Proteins **3**: 71-84.
11. Frishman, D. and P. Argos (1995). Knowledge-based protein secondary structure assignment. Proteins **23**: 566-579.
12. Cuff, J. A. and G. J. Barton (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins **34**: 508-519.
13. 郝柏林、刘寄星，理论物理与生命科学，上海：上海科学技术出版社，1997
14. Claverie, J. M. (2000). From bioinformatics to computational biology. Genome Res **10**: 1277-1279.

15. M. O. Dayhoff, R. M. Schwartz, B. C. Orcutt, A model of evolutionary change in proteins, in Atlas of Protein Sequence and Structure, ed. by M. O. Dayhoff, Washington DC, National Biomedical Research Foundation, 1978, 345-352; R. M. Schwartz, M. O. Dayhoff, “Matrices for detecting distant relationships”, 见同书 353-358
16. Henikoff, S. and J. G. Henikoff (1992). Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A **89**: 10915-10919.
17. Gibbs, A. J. and G. A. McIntyre (1970). The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences. Eur J Biochem **16**: 1-11.
18. R. Bellman, Dynamic Programming, American: Princeton Press, 1957
19. Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol **48**: 443-453.
20. Smith, T. F. and M. S. Waterman (1981). Identification of common molecular subsequences. J Mol Biol **147**: 195-197.
21. Lipman, D. J. and W. R. Pearson (1985). Rapid and sensitive protein similarity searches. Science **227**: 1435-1441.
22. Pearson, W. R. and D. J. Lipman (1988). Improved tools for biological sequence comparison. Proc Natl Acad Sci U S A **85**: 2444-2448.
23. Altschul, S. F., W. Gish, et al. (1990). Basic local alignment search tool. J Mol Biol **215**: 403-410.
24. Altschul, S. F., T. L. Madden, et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res **25**: 3389-3402.
25. Thompson, J. D., D. G. Higgins, et al. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res **22**: 4673-4680.
26. Notredame, C., D. G. Higgins, et al. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol **302**: 205-217.
27. Karlin, S. and S. F. Altschul (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc Natl Acad Sci U S A **87**: 2264-2268.

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

29. Tatusov, R. L., E. V. Koonin, et al. (1997). A genomic perspective on protein families. Science **278**: 631-637.

30. Tatusov, R. L., D. A. Natale, et al. (2001). The COG database: new developments in phylogenetic classification of proteins from complete genomes. Nucleic Acids Res **29**: 22-28.

31. Borodovsky, M. and J. McIninch (1993). GENMARK: parallel gene recognition for both DNA strands. Comput Chem **17**: 123-133.

32. Lukashin, A. V. and M. Borodovsky (1998). GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res **26**: 1107-1115.

33. Salzberg, S. L., A. L. Delcher, et al. (1998). Microbial gene identification using interpolated Markov models. Nucleic Acids Res **26**: 544-548.

34. Delcher, A. L., D. Harmon, et al. (1999). Improved microbial gene identification with GLIMMER. Nucleic Acids Res **27**: 4636-4641.

35. Majoros, W. H., M. Pertea, et al. (2003). GlimmerM, Exonomy and Unveil: three ab initio eukaryotic genefinders. Nucleic Acids Res **31**: 3601-3604.

36. Besemer, J., A. Lomsadze, et al. (2001). GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. Nucleic Acids Res **29**: 2607-2618.

37. Audic, S. and J. M. Claverie (1998). Self-identification of protein-coding regions in microbial genomes. Proc Natl Acad Sci U S A **95**: 10026-10031.

38. Cardon, L. R. and G. D. Stormo (1992). Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. J Mol Biol **223**: 159-170.

39. Tiwari, S., S. Ramachandran, et al. (1997). Prediction of probable genes by Fourier analysis of genomic sequences. Comput Appl Biosci **13**: 263-270.

40. Yan, M., Z. S. Lin, et al. (1998). A new fourier transform approach for protein coding measure based on the format of the Z curve. Bioinformatics **14**: 685-690.

41. Fukushima, A., T. Ikemura, et al. (2002). Periodicity in prokaryotic and eukaryotic genomes identified by power spectrum analysis. Gene **300**: 203-211.

42. Stephane, Mallat, A Wavelet Tour of Signal Processing (Second Edition) （杨力华等译），北京：机械工业出版社，2002
43. Tsonis, A. A., P. Kumar, et al. (1996). Wavelet analysis of DNA sequences. Physical Review. E. Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **53**: 1828-1834.
44. Lio, P. and M. Vannucci (2000). Finding pathogenicity islands and gene transfer events in genome data. Bioinformatics **16**: 932-940.
45. Wen, S. Y. and C. T. Zhang (2003). Identification of isochore boundaries in the human genome using the technique of wavelet multiresolution analysis. Biochem Biophys Res Commun **311**: 215-222.
46. Lio, P. (2003). Wavelets in bioinformatics and computational biology: state of art and perspectives. Bioinformatics **19**: 2-9.
47. Hamori, E. and Ruskin, J. (1983). H curves, a novel method of representation of nucleotide series especially suited for long DNA sequences. J Biol Chem **258**: 1318-1327.
48. Zhang, C. T. and R. Zhang (1991). Analysis of distribution of bases in the coding sequences by a diagrammatic technique. Nucleic Acids Res **19**: 6313-6317.
49. Zhang, R. and C. T. Zhang (1994). Z curves, an intutive tool for visualizing and analyzing the DNA sequences. J Biomol Struct Dyn **11**: 767-782.
50. Zhang, C. T., Z. S. Lin, et al. (1998). A novel approach to distinguish between intron-containing and intronless genes based on the format of Z curves. J Theor Biol **192**: 467-473.
51. Zhang, C. T. and J. Wang (2000). Recognition of protein coding genes in the yeast genome at better than 95% accuracy based on the Z curve. Nucleic Acids Res **28**: 2804-2814.
52. Guo, F. B., H. Y. Ou, et al. (2003). ZCURVE: a new system for recognizing protein-coding genes in bacterial and archaeal genomes. Nucleic Acids Res **31**: 1780-1789.
53. Chen, L. L., H. Y. Ou, et al. (2003). ZCURVE_CoV: a new system to recognize protein coding genes in coronavirus genomes, and its applications in analyzing SARS-CoV genomes. Biochem Biophys Res Commun **307**: 382-388.
54. 谢惠民，复杂性与动力系统，上海：上海科技教育出版社，1994
55. 杨光正、吴岷、张晓莉，模式识别，合肥：中国科学技术出版社，2001

56. Dong, S. and D. B. Searls (1994). Gene structure prediction by linguistic methods. Genomics **23**: 540-551.

57. Pesole, G., M. Attimonelli, et al. (1996). Linguistic analysis of nucleotide sequences: algorithms for pattern recognition and analysis of codon strategy. Methods Enzymol **266**: 281-294.

58. Searls, D. B. (1997). Linguistic approaches to biological sequences. Comput Appl Biosci **13**: 333-344.

59. Cyntbia Gibas and Per Jambeck, Bioinformatics Computer Skills, 北京：科学出版社（英文影印版），2002

60. 史忠植，知识发现，北京：清华大学出版社，2002

61. Salzberg, S. (1995). Locating protein coding regions in human DNA using a decision tree algorithm. J Comput Biol **2**: 473-485.

62. Salzberg, S., A. L. Delcher, et al. (1998). A decision tree system for finding genes in DNA. J Comput Biol **5**: 667-680.

63. Selbig, J., T. Mevissen, et al. (1999). Decision tree-based formation of consensus protein secondary structure prediction. Bioinformatics **15**: 1039-1046.

64. Dan Gusfield , Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, UK: Cambridge Univ Pr (Short) , 1997)

65. Delcher, A. L., S. Kasif, et al. (1999). Alignment of whole genomes. Nucleic Acids Res **27**: 2369-2376.

66. Delcher, A. L., A. Phillippy, et al. (2002). Fast algorithms for large-scale genome alignment and comparison. Nucleic Acids Res **30**: 2478-2483.

67. Marsan, L. and M. F. Sagot (2000). Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. J Comput Biol **7**: 345-362.

68. Gracy, J. and P. Argos (1998). Automated protein sequence database classification. I. Integration of compositional similarity search, local similarity search, and multiple sequence alignment. Bioinformatics **14**: 164-173.

69. 任若恩、王惠文，多元统计数据分析——理论、方法、实例，北京：国防工业出版社，1997

70. Zhang, M. Q. (1997). Identification of protein coding regions in the human genome by quadratic discriminant analysis. Proc Natl Acad Sci U S A **94**: 565-568.

71. Zhang, M. Q. (2000). Discriminant analysis and its application in DNA sequence motif recognition. Brief Bioinform **1**: 331-342.

72. von Heijne, G. (1986). A new method for predicting signal sequence cleavage sites. Nucleic Acids Res **14**: 4683-4690.

73. Zhang, M. Q. and T. G. Marr (1993). A weight array method for splicing signal analysis. Comput Appl Biosci **9**: 499-509.

74. Bai-lin Hao (2000) Fractals from genomes: exact solutions of a biology-inspired problem, Physica A 282: 225-246

75. B. L. Hao, H. C. Lee and S. Y. Zhang (2000) Fractals related to long DNA sequences and complete genomes, Chaos, Solitons and Fractals, **11:** 825-836

76. P. Tino (2002) Multifractal properties of Hao's geometric representation of DNA sequences, Physica A **304:** 480-494

77. Jeffrey, H. J. (1990). Chaos game representation of gene structure. Nucleic Acids Res **18**: 2163-2170.

78. Fiser, A., G. E. Tusnady, et al. (1994). Chaos game representation of protein structures. J Mol Graph **12**: 302-4, 295.

79. Almeida, J. S., J. A. Carrico, et al. (2001). Analysis of genomic sequences by Chaos Game Representation. Bioinformatics **17**: 429-437.

80. Almeida, J. S. and S. Vinga (2002). Universal sequence map (USM) of arbitrary discrete sequences. BMC Bioinformatics **3**: 6.

81. Mohanty, A. K. and A. V. Narayana Rao (2000). Factorial moments analyses show a characteristic length scale in DNA sequences. Phys Rev Lett **84**: 1832-1835.

82. Bernaola-Galvan, P. and P. Carpena (2002). Comment on "Factorial moments analyses show a characteristic length scale in DNA sequences". Phys Rev Lett **88**: 219803; discussion 219804.

83. Rost, B. (2001). Review: protein secondary structure prediction continues to rise. J Struct Biol **134**: 204-218.

84. Chou, P,Y. and Fasman, G. D (1974) Prediction of protein conformation. Biochemistry, **13**: 222-244.

85. Lim, V. I. (1974). Algorithms for prediction of alpha-helical and beta-structural regions in globular proteins. J Mol Biol **88**: 873-894.

86. Garnier, J., D. J. Osguthorpe, et al. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J. Mol. Biol. **120**: 97-120.

87. Cohen, F. E., R. M. Abarbanel, et al. (1986). Turn prediction in proteins using a pattern-matching approach. Biochemistry **25**: 266-275.

88. Qian, N. and T. J. Sejnowski (1988). Predicting the secondary structure of globular proteins using neural network models. J Mol Biol **202**: 865-884.

89. Cuff, J. A. and G. J. Barton (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Proteins **40**: 502-511.

90. Rost, B. and C. Sander (1993). Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol **232**: 584-599.

91. Petersen, T. N., C. Lundegaard, et al. (2000). Prediction of protein secondary structure at 80% accuracy. Proteins **41**: 17-20.

92. Pollastri, G., D. Przybylski, et al. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins **47**: 228-235.

93. Barton, G. J. (1995). Protein secondary structure prediction. Curr Opin Struct Biol **5**: 372-376.

94. Baldi, P., S. Brunak, et al. (1999). Exploiting the past and the future in protein secondary structure prediction. Bioinformatics **15**: 937-946.

95. Cuff, J. A. and G. J. Barton (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins **34**: 508-519.

96. Rost, B., C. Sander, et al. (1994). Redefining the goals of protein secondary structure prediction. J Mol Biol **235**: 13-26.

97. Zemla, A., C. Venclovas, et al. (1999). A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. Proteins **34**: 220-223.

98. Zhang, C. T. and R. Zhang (2001). A refined accuracy index to evaluate algorithms of protein secondary structure prediction. Proteins **43**: 520-522.

99. Zhang, C. T. and R. Zhang (2003). Q9, a content-balancing accuracy index to evaluate algorithms of protein secondary structure prediction. Int J Biochem Cell Biol **35**: 1256-1262.

100. Zhang, C. T. and K. C. Chou (1992). An optimization approach to predicting protein structural class from amino acid composition. Protein Sci **1**: 401-408.

101. 杨建刚，人工神经网络实用教程，杭州：浙江大学出版社，2001

102. Salamov, A. A. and V. V. Solovyev (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. J Mol Biol **247**: 11-15.

103. Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol **292**: 195-202.

104. Ferran, E. A., B. Pflugfelder, et al. (1994). Self-organized neural maps of human protein sequences. Protein Sci **3**: 507-521.

105. Vladimir N. Vapnik, 统计学习理论的本质 (张学工)译，北京：清华大学出版社，2000

106. 边肇祺、张学工，模式识别（第二版），北京：清华大学出版社，2000

107. Hua, S. and Z. Sun (2001). A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol **308**: 397-407.

108. Cai, C. Z., W. L. Wang, et al. (2003). Protein function classification via support vector machine approach. Math Biosci **185**: 111-122.

109. Cai, C. Z., L. Y. Han, et al. (2003). SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res **31**: 3692-3697.

110. Donaldson, I., J. Martin, et al. (2003). PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics **4**: 11.

111. Bao, L. and Z. Sun (2002). Identifying genes related to drug anticancer mechanisms using support vector machine. FEBS Lett **521**: 109-114.

112. Zien, A., G. Ratsch, et al. (2000). Engineering support vector machine kernels that recognize translation initiation sites. Bioinformatics **16**: 799-807.

113. Sun, Y. F., X. D. Fan, et al. (2003). Identifying splicing sites in eukaryotic RNA: support vector machine approach. Comput Biol Med **33**: 17-29.

114. Furey, T. S., N. Cristianini, et al. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics **16**: 906-914.

115. 中国大百科全书数学编辑委员会，中国大百科全书（数学卷），北京：中国大百科全书出版社，1988

116. Stormo, G. D., T. D. Schneider, et al. (1986). Quantitative analysis of the relationship between nucleotide sequence and functional activity. Nucleic Acids Res **14**: 6661-6679.

117. Barrick, D., K. Villanueba, et al. (1994). Quantitative analysis of ribosome binding sites in E.coli. Nucleic Acids Res **22**: 1287-1295.

118. Zhang, C. T., Z. S. Lin, et al. (1998). Prediction of the helix/strand content of globular proteins based on their primary sequences. Protein Eng **11**: 971-979.

119. Pan, X. M. (2001). Multiple linear regression for protein secondary structure prediction. Proteins **43**: 256-259.

120. Horimoto, K. and H. Toh (2001). Statistical estimation of cluster boundaries in gene expression profile data. Bioinformatics **17**: 1143-1151.

121. R. Durbin, S. Eddy and A. Krogh etc., Biological sequence analysis: probabilistic models of proteins and nucleic acids, 北京：清华大学出版社（英文引进版），2002

122. Baldi P. and Brunck S., Bioinformatics: the machine learning approach, American: MIT press, Cambridge, Massachusetts, 1998

123. 潘承洞、潘承彪，解析数论基础，北京：科学出版社，1991

124. Delamarche, C., Guerdoux-Jamet, P. and Gras, R. *et al*. (1999) A symbolic-numeric approach to find patterns in genomes. Application to the translation initiation sites of E. coli. Biochimie, **81:** 1065-1072

125. 张立昂，可计算性与计算复杂性导引，北京：北京大学出版社，1996

126. 王树恩、陈士俊，科学技术论与科学技术创新方法论，天津：南开大学出版社，2001

127. 徐利治、王仁宏、周蕴时，函数逼近的理论与方法，上海：上海科学技术出版社，1983

128. 李建平、唐远炎，小波分析方法的应用，重庆：重庆大学出版社，1999

129. 李建平，小波分析与信号处理——理论、应用及软件实现，重庆：重庆大学出版社，1997

130. Stephane, Mallat, A Wavelet Tour of Signal Processing (Second Edition) （杨力华等译），北京：机械工业出版社，2002

131. 赵松年、熊小芸，子波变换与子波分析，北京：电子工业出版社，1996

132. 程云鹏，矩阵论（第二版），西安：西北工业大学出版社，2000

133. Tufte, E. R. (1983) The Visual Display of Quantitative Information, Graphics Press, Cheshire, Connecticut.

134. Schneider, T. D. and R. M. Stephens (1990). Sequence logos: a new way to display consensus sequences. Nucleic Acids Res **18**: 6097-6100.

135. Mathe C., Sagot M.F., Schiex T. et al., Current methods of gene prediction, their strengths and weaknesses, Nucleic Acids Res, 2002, **30**: 4103-4117

136. Frishman D., Mironov A., Mewes H.W. et al., Combining diverse evidence for gene recognition in completely sequenced bacterial genomes, Nucleic Acids Res, 1998, **26**: 2941-2947

137. Badger J.H., Olsen G.J., CRITICA: coding region identification tool invoking comparative analysis, Mol Biol Evol, 1999, **16**: 512-524

138. Shepherd J.C., Method to determine the reading frame of a protein from the purine/pyrimidine genome sequence and its possible evolutionary justification, Proc Natl Acad Sci U S A, 1981, **78**: 1596-1600

139. Fickett J.W., Recognition of protein coding regions in DNA sequences, Nucleic Acids Res, 1982, **10**: 5303-5318

140. Staden R., McLachlan A.D., Codon preference and its use in identifying protein coding regions in long DNA sequences, Nucleic Acids Res, 1982, **10**: 141-156

141. Tsonis A.A., Elsner J.B., Tsonis P.A., Periodicity in DNA coding sequences: implications in gene evolution, J Theor Biol, 1991, **151**: 323-331

142. Bibb M.J., Findlay P.R., Johnson M.W., The relationship between base composition and codon usage in bacterial genes and its use for the simple and reliable identification of protein-coding sequences, Gene, 1984, **30**: 157-166

143. Fickett, J. W. and C. S. Tung (1992). Assessment of protein coding measures. Nucleic Acids Res **20**: 6441-6450.

144. Yada T., Totoki Y., Takagi T. et al., A novel bacterial gene-finding system with improved accuracy in locating start codons, DNA Res, 2001, **8**: 97-106

145. Guigo R., Knudsen S., Drake N. et al., Prediction of gene structure, J Mol Biol, 1992, **226**: 141-157

146. Burge C., Karlin S., Prediction of complete gene structures in human genomic DNA, J Mol Biol, 1997, **268**: 78-94

147. Stanke M., Waack S., Gene prediction with a hidden Markov model and a new intron submodel, Bioinformatics, 2003, 19 Suppl 2: II215-II225

148. Stormo G.D., Gene-finding approaches for eukaryotes, Genome Res, 2000, 10: 394-397

149. Shibuya T., Rigoutsos I., Dictionary-driven prokaryotic gene finding, Nucleic Acids Res, 2002, **30**: 2710-2725

150. Burset, M. and R. Guigo (1996). Evaluation of gene structure prediction programs. Genomics **34**: 353-367.

151. Fickett, J. W. (1996). The gene identification problem: an overview for developers. Comput chem **20**: 103-118.

152. Guigo, R. (1997). Computational gene identification: an open problem. Comput Chem **21**: 215-222.

153. Guigo, R., P. Agarwal, et al. (2000). An assessment of gene prediction accuracy in large DNA sequences. Genome Res **10**: 1631-1642.

154. Bajic, V. B. (2000). Comparing the success of different prediction software in sequence analysis: a review. Brief Bioinform **1**: 214-228.

155. Rogic, S., A. K. Mackworth, et al. (2001). Evaluation of gene-finding programs on mammalian sequences. Genome Res **11**: 817-832.

发表论文和参加科研情况说明

在攻读硕士的这两年多的时间里，作者主要致力于字符序列的解析数论模型的研究。期间，共写了三篇英文论文和一篇中文论文。前两篇英文论文是关于对偶描述子理论的初期想法的，第三篇英文论文是关于蛋白质编码区碱基组成的相位特异性与功能关系的。这三篇论文皆因水平不够而未获准发表。那一篇中文论文是关于原核生物基因组中负样本的生成的（参见附录 III），投到了天津理工学院的学报上。

作者的主要研究成果就表现为写在本毕业论文中的字符序列解析数论模型的思想、方法和它的面向对象的实现：DD 类。此外，作者还对两类预测算法的评价问题做了一点研究工作。

发表的论文就是那篇中文的：

马彬广，（2004），原核基因识别中的一种负样本生成算法，天津理工学院学报，第 20 卷，总第 65 期：

附录 I：符号说明

集合与元素

C	字符集；
c_i	字符集 C 中的第 i 个元素；
C^*	由字符集 C 中的元素所构成的字符串的集合；
X 或 $\mathbf{X}$	组成权重因子的集合，$X \subset R-\{0\}$，$\mathbf{X} \subset R^m-\{\mathbf{0}\}$；
x_i 或 $\mathbf{x}_i$	第 i 个组成权重因子，$x_i \in X$，$\mathbf{x}_i \in \mathbf{X}$；
R 和 R^m	分别表示实数集和 m 维实空间；

序列与元素

s	由字符集 C 中的元素所构成的字符串，$s \in C^*$；
x 或 $\mathbf{x}$	由 X 或 $\mathbf{X}$ 中的元素所构成的实数（或实矢量）的序列；
$x[k]$ 或 $\mathbf{x}[k]$	序列 x 或 $\mathbf{x}$ 中的第 k 个元素，有时也表示序列，即等同于 x 或 $\mathbf{x}$，由上下文确定；
$\mathbf{s}$	对偶矢量的序列，即，$\mathbf{x}$ 的加权和序列；
s'	表示待识别序列或重构出来的字符序列（看上下文）；
x'	表示重构出来的 x；

函数与算符

$\lfloor \square \rfloor$	不大于“$\sqcup$”的最大整数；
$\lceil \square \rceil$	不小于“$\sqcup$”的最小整数；
$\delta(\square)$	当 $\square = 0$ 时，它等于 1；当 $\square \neq 0$ 时，它等于 0；
arcmin()	min() 函数的反函数；
mod	取模运算符；
$\rightarrow$	映射符，$f: C \rightarrow X$ 表示 C 到 X 的映射，$\mathrm{A} \rightarrow 1$ 表示把字符 A 映射成实数 1；

附录 II： 序列扩增的两种方法

在字符序列处理的问题中，有时候需要把短序列延长，把少量的序列变多，这就是序列的扩增。比如，在原核生物基因识别问题中，有时候，选择基因间序列作为参考序列。由于原核基因组中基因间序列都是短片段，而且量很少，如果大量需要，就得想办法扩增。下面介绍两种序列扩增方法。

以 DNA 序列为例，设有 5 个 DNA 序列的片段，分别为：ACGT、TCAG、AATGC、TGA、GCTA，要求用上面的这些片段，构造一条长度为 1000 的 DNA 序列，可以使用下面的两种方法之一。

方法 1：横向延拓法。 把这些序列片段首尾相接，串联起来，可以得到一条长度为 20 的 DNA 序列 ACGTTCAGAATGCTGAGCTA，然后，继续使用这些片段，再往头尾上接，如此反复，直到长度达到要求为止。这种方法会人为引入周期性，但保留了短片段的局部特征。如果不想保留这种局部特征，则可以将所得的 DNA 序列再随机打乱。这样一来，人为引入的周期性就不复存在了。这种方法曾有人使用过[40]。

方法 2：纵向扩展法。构造“自相似映射”，比如：A→TCAG，C→AATGC，G→TGA，T→GCTA，然后，任选一个片段作为种子，比如选 ACGT，然后，把其中的 A 用 TCAG 代替，把其中的 C 用 AATGC 代替，其中的 G 用 TGA 代替，其中的 T 用 GCTA 代替，就得到字符串 TCAGAATGCTGAGCTA。接下来，再把字符串 TCAGAATGCTGAGCTA 中的 A、C、G、T 分别用 TCAG、AATGC、TGA、GCTA 来代替。如此重复下去，直至序列的长度达到要求为止。这种方法也会人为引入一定的周期性，但没有方法 1 那样明显。同时，它也保留了短片段的局部特征。如果不想保留这种局部特征，则可以将所得的 DNA 序列再随机打乱。这样一来，人为引入的周期性就不复存在了。

无论是“横向延拓法”还是“纵向扩展法”，所直接得到的序列，都保留了原来短序列片段的局部特征，所以是和原片段相似的序列。这就达到了把序列由短变长，从少到多的目的。如果只想生成的序列在字符组成上，和原序列片段保持相似性，而不关心排列方式，则可将所得到的序列再进一步随机打乱，具体做法是：生成 1 到序列长度之间均匀分布的两个随机数，把这两个随机数所对应的位置处的两个字符，交换一下，重复这一过程多次后，序列就被打乱了。

致谢

首先感谢导师张春霆院士。与张老师的每次谈话，都令我很受教益。我能取得今天的一点小小成绩，是与张老师的“放手与监督”相结合的培养方案分不开的。由于“放手”，使我获得了充分的自由支配的时间，来发展自己的兴趣；由于“监督”，保证了我科研的高起点与正确方向。相信张老师严父般的关爱，必将使我终身受益。

其次，我要感谢曾在天津大学生物信息学实验室工作的王永宏、黄积涛、王举和冯志萍等各位老师，他们在学习和工作中都曾给予我指点和帮助。感谢欧竑宇、郭锋彪、陈玲玲、杨晓光等各位师兄师姐，他们在科研上教会我很多具体的技巧，在生活上给过我热心的关照。感谢和我一届入学的高峰和温省云同学，在一起求学的道路上，他们曾给我很多帮助。我会铭记这段如切如磋、情若手足的同窗之谊。感谢窦运涛、郑文新和张建辉等师弟师妹，与他们一起讨论学习、共同进步，曾带给我很大的乐趣。感谢曾经在实验室待过的王明涛、罗瑞燕、张娜和魏恒等同学，他们优异的成绩和远大的志向，都曾是激励我前进的动力。还要特别感谢董传伟同学，在处理计算机问题时，在一起合作带的物理实验课上，他都曾给过我热心的帮助。

感谢室友褚效中同学对我的学习和生活的理解、支持和帮助。

父母的养育之恩，是我毕生都难以报答的。一起长大的妹妹，则一直关心和支持着我。这种亲情难以言表。只能以这篇论文献给他们，让他们感到些许的欣慰。

还有很多老师、同学、亲人和朋友对我的学习、工作和生活曾经给予过并还在给予着关怀和帮助，这里虽然没有提到他们的名字，但作者一样由衷地向他们表示最真诚的感谢。

最后，向各参考文献的作者表示感谢。