基于生物信息学特征的 DNA 序列数据压缩算法

纪 震[1],周家锐[2],朱泽轩[1],Q H Wu[3]
(1.深圳大学计算机与软件学院,广东深圳 518060; 2.浙江大学生物医学工程与仪器科学院,浙江杭州 310027;
3.利物浦大学电气电子工程系,利物浦,L69 3GJ,UK)

摘 要: 本文通过将生物学特征和生物学含义引入 DNA 序列数据的压缩处理中,提出了基于生物信息学特征的 BioLZMA 压缩算法.在 BioLZMA 算法中,DNA 序列根据组成部分生物学含义的不同切分重组为四个集合:编码序列 CDS 集合、内含子序列集合、RNA 序列集合以及剩余序列的集合.根据各集合中序列的具体生物学特征分别使用针对性的压缩策略进行预处理,并通过 LZMA 算法进行压缩编码.实验结果表明,BioLZMA 算法在基准测试序列上的压缩性能优于原有的 DNA 序列压缩方法.特别是对于生物信息学特征清晰的长序列,算法能够在较短的时间内获得较高的压缩率.

关键词: DNA 数据压缩; 生物信息学; 序列重组; 近似重复片段; LZMA
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2011) 05-0991-05

Bioinformatics Features Based DNA Sequence Data Compression Algorithm

JI Zhen[1], ZHOU Jia-rui[2], ZHU Ze-xuan[1], Q H Wu[3]
(1. *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong* 518060, *China*;
2. *College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou, Zhejiang* 310027, *China*;
3. *Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, L69 3GJ, UK*)

Abstract: A novel bioinformatics features based DNA Sequence data compression algorithm of BioLZMA is proposed in this paper. In BioLZMA, the DNA sequence data is sliced and reformed into 4 clusters according with biological meanings: the coding sequence cluster, the intron cluster, the RNA cluster and the residual cluster. By employing pointed compression strategies in data pre-processing, the clusters are compressed separately with LZMA algorithm. Experimental results demonstrated the better performance of BioLZMA than original DNA compression algorithms on benchmark sequences. Especially on long DNA sequence with significant bioinformatics features, BioLZMA algorithm can achieve higher compression ratio with little computation time.

Key words: DNA sequence data compression; bioinformatics; sequence regroup; approximate repeat fragment; Lempel-Ziv-Markov chain algorithm (LZMA)

1 引言

DNA 序列数据广泛应用于生物学、医学、遗传科学等诸多领域,具有重要研究价值.用以获取 DNA 序列基础数据的生物测序工程近年来得到了广泛重视.随着各种测序项目的展开,产生的序列数据量呈指数规模增长[1],平均约每 14 个月增加一倍.如何以更低的成本处理快速积累的 DNA 信息,在满足相关研究需求的同时有效减轻大量数据带来的存储与传输压力,是一项重要而紧迫的课题.以高效的压缩编码方法,将 DNA 序列数据存储于较小的空间中成为必然的选择.但由于 DNA 数据构成的特殊性,传统压缩算法效果并不理想,甚至出现处理后数据量反而有所增长的情况[2].从而催生了针对 DNA 序列数据的专用压缩算法.

1993 年 S Grumbach 和 F Tahi 首次提出了 BioCompress 算法[3],通过在 LZ 算法中加入对互补回文结构(Complementary palindromes)的压缩编码,有效降低了序列冗余度.尽管 BioCompress 算法总体性能并不理想,但与传统方法相比在 DNA 序列数据上的压缩率仍有较大提升.1999 年 X Chen 和 S Kwong 等提出的 GenCompress 算法是 BioCompress 算法的改进[4].通过添加对近似重复序列的编码处理,有效提高了压缩率.2000 年 T Matsumoto 和 K Sadakane 提出了 CTW + LZ 算法[5],将上下文树加权(Context tree weighting, CTW)算法和 LZ 压缩方

收稿日期:2009-12-16;修回日期:2010-02-24
基金项目:国家自然科学基金(No.60872125);霍英东教育基金会高等院校青年教师基金基础性研究课题;深圳市基础研究项目(杰青奖);广东省自然科学基金

法相结合,使用多个编码模型对 DNA 序列的不同片断进行压缩.尽管 CTW + LZ 算法在压缩率上有所提升,但其压缩速度下降明显.2002 年 X Chen 和 M Li 等提出了 DNACompress 压缩算法[6],使用 PatternHunter 工具搜索 DNA 序列的重复与近似重复片断,提高了算法的整体速度.2007 年 G Korodi 和 I Tabus 提出了 GeNML 算法[7],基于归一化最大似然(Normalized maximum likelihood, NML)模型对 DNA 序列数据进行压缩.GeNML 算法对具有不同数据特点的 DNA 片断使用不同的编码策略和概率模型进行压缩,获得了较好的性能.

原有 DNA 序列压缩算法仅考虑了数据的构成特点,仅将序列视作具有特殊构成特点的长字符串进行整体处理,而没有考虑各组成片段的具体生物学信息,其性能提升已逐渐陷入瓶颈[8].为解决这一问题,我们将 DNA 序列的生物信息学特征应用于压缩处理中,提出了 BioLZMA 压缩算法.通过将序列的生物学含义与生物学特征引入数据预处理,配合以高性能的 LZMA 压缩技术,BioLZMA 算法能够在较短的时间内获得比传统方法更高的压缩率.

2 DNA 序列数据的生物信息学特征

生物信息学是利用应用数学、信息学、统计学和计算机科学的方法研究生物学问题的一门交叉学科[9],使用各种数学工具从 DNA 序列中提取有效的生物学信息,以对其进行搜索、处理与应用.通过将 DNA 数据的生物信息学特征引入压缩算法,挖掘并利用序列中的生物学信息,将能有效提升算法性能.在 BioLZMA 算法中,使用的生物信息学特征包括生物学含义和生物学特征两方面.

首先,DNA 序列的不同区域有着不同的生物学含义,其功能不同,碱基分布特点也有所差异.如图 1 所示,DNA 序列按其最小生物功能可划分为外显子(Exon)、内含子(Intron)、RNA 片段和基因间区段四个部分[10].其中,外显子是基因中直接编码为多肽链的片断,具有冗余度低、生物学含义明晰的特点.基因中单个或多个外显子片断可链接组成编码区域(Coding sequences, CDS),通过一定阅读规则表达为氨基酸序列.内含子是基因中各外显子片断间的部分,在翻译过程中将被舍弃.RNA 片段主要包括 tRNA 和 rRNA,是序列中用以辅助表达的部分,其碱基排列较为保守,片段间相似度较高.基因间区段用于构成 DNA 序列的结构,往往不含生物学信息,因而碱基排列较为不规则.但由于其与 DNA 的物质性质直接相关,碱基的出现具有显著的规律性.

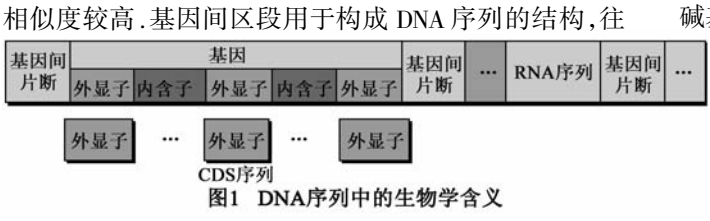


图1 DNA序列中的生物学含义

在压缩算法中,通过将 DNA 序列中具有不同功能的片断进行切分重组,可使其生物学含义明晰化,有效提升各部分内片段的相似度,从而改进处理性能.根据序列各组成片段生物学信息差异使用针对性的压缩策略,是本算法与传统方法最大的区别之一.而 DNA 数据的功能划分可直接由序列的注释信息获得.

其次,DNA 序列数据含有一定的生物学特征,具有相似性特点.但与一般数据不同,DNA 序列的重复有其独有特点,需要进行针对性的搜索处理.

第一,DNA 序列中含有大量重复片断.DNA 数据中的重复部分约占序列总量的 32%,不同 DNA 序列间也具有一定相似性.例如,同属灵长目的黑猩猩与人类,其 DNA 序列的相似性可达到 96%[11].通过对序列的重复部分进行编码索引,可有效提升算法压缩率.

第二,DNA 序列中的重复具有多种模式.如图 2 所示,除常见的直接重复外,DNA 序列还具有镜像重复、反转重复、配对重复和互补回文结构等特有模式.尽管反转重复和互补回文结构表现在 DNA 的双链上,但根据碱基配对准则,这两种模式可视为在单链上由中心点向两端碱基逐个配对的形式.在针对 DNA 序列数据的压缩算法中,这些重复模式都需要进行针对性的处理.

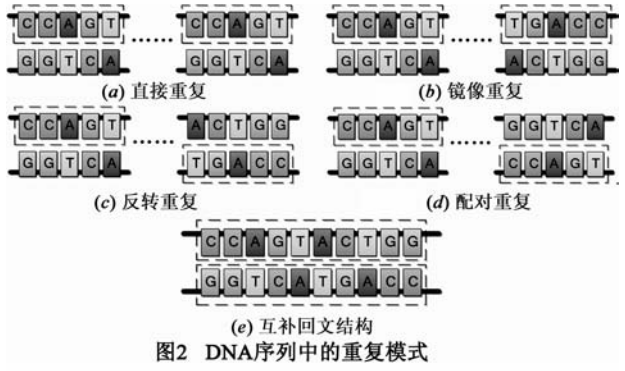


图2 DNA序列中的重复模式

第三,DNA 序列中的重复更多的表现为近似重复形式.上述重复模式在实际 DNA 序列中的出现往往含有一定的误差.即在相似片断的基础上,可通过插入、删除或替换有限个碱基转化为精确重复的序列.在压缩算法中,可通过添加附加信息的方式将近似重复片断变为精确重复片断进行压缩编码.但必须控制误差碱基的个数,以避免添加过多的附加信息导致修正后序列压缩率反而有所膨胀.

在 BioLZMA 算法中,上述的生物信息学特征都被应用于 DNA 序列数据的压缩处理中,以取得提升压缩性能的目的.

3　BioLZMA 压缩算法

在 BioLZMA 算法中,首先根据序列的生物信息学特征对 DNA 数据进行预处理,使其分布更具有规律性.而后将处理后的序列和附加信息编码为二进制数据流,分别通过高效的 LZMA 算法进行压缩运算.

在预处理阶段,首先根据生物学含义的不同将 DNA 序列切分重组为多个集合:CDS 集合包含了基因内的外显子片断;Intron 集合包含了基因内的内含子片断;RNA 集合包含了序列内的 RNA 片段;Other 集合包含了以基因间区段为主的其他剩余部分.BioLZMA 算法同时记录了各集合中组成片断在原始序列的位置与长度,以便解压缩时使用.

对于 CDS 集合,算法采取将外显子 DNA 数据根据其实际阅读规则,编码为氨基酸序列后进行压缩的方法.标准氨基酸包括终止符号共 23 种,若直接进行二进制编码,则每个氨基酸符号需要 5bit 空间.但不同种类氨基酸在生物体内的出现概率各不相同,一些氨基酸如硒半胱胺酸等仅在某些特定序列中有极少量分布.通过对氨基酸符号使用 Huffman 编码,能够有效减少输出数据的长度.常见氨基酸的出现概率及其 Huffman 二进制编码如表 1 所示[12].另外,由于同一氨基酸符号可能对应多种 DNA 组合,编码时需要添加附加数据位表示原始碱基组合的信息.

表 1　常见氨基酸出现概率及编码

氨基酸	出现概率(%)	编码	氨基酸	出现概率(%)	编码
Ala	7.8	1101	Met	2.3	11111
Cys	1.9	100111	Asn	4.3	11110
Asp	5.3	0100	Pro	5.2	0011
Glu	6.3	1000	Gln	4.2	11101
Phe	3.9	11100	Arg	5.1	0010
Gly	7.2	1100	Ser	6.8	1011
His	2.3	111110	Thr	5.9	0111
Ile	5.3	0101	Val	6.6	1010
Lys	5.9	0110	Trp	1.4	100110
Leu	9.1	000	Tyr	3.2	10010

对于另外三个集合,采用分别压缩其 DNA 序列数据的方法.对各集合,首先进行精确重复片断与近似重复片断的搜索,包括全部四种重复模式.再对找到的重复序列进行整合,挑选其中范围不重叠的最长片断作为匹配结果.而后对 DNA 序列进行修正,使所有匹配片段都转化为直接精确重复的形式.在生成附加信息时,对于精确重复的片断只需记录其重复模式和数据长度即可,而对于近似重复片断则需另外记录每个修正碱基符号的位置与原始状况信息.最后,将修正后的 DNA 序列和附加信息编码为二进制数据,每个碱基符号使用 2bit 数据表示.

在压缩阶段,BioLZMA 使用 LZMA 算法对序列数据进行压缩处理.LZMA (Lempel-Ziv-Markov chain algorithm) 压缩算法于 2001 年由 Igor Pavlov 提出[13],是 LZ77 算法的有效改进.LZMA 算法使用散列表、二叉树和基数树等结构对序列进行字典查找,对数据流、重复序列大小以及重续序列位置单独进行压缩.通过引入独有的 Range Coder 熵编码算法和 Price 机制,LZMA 算法能够在较短的时间内获得较高的压缩率,具有重要的实用价值.

压缩时,BioLZMA 算法将 CDS 集合的氨基酸序列编码二进制数据和另外三个集合的修正 DNA 序列二进制数据分别输入 LZMA 算法进行压缩运算.选取字典大小为 128MB,单词长度为 64.压缩后若输出数据长度大于原始序列长度,则使用原始序列表示,否则使用压缩后数据表示.实验表明,仅在压缩 RNA 集合时会有极少数几率出现数据长度增加的情况,因此一般不会对压缩性能产生太大影响.BioLZMA 算法流程如表 2 所示.

表 2　BioLZMA 算法流程

Step 1	对 DNA 序列 $v=\{v_1,v_2,\cdots\}$,根据各片段的生物学含义切分重组为四个子部分 $v=\{v_{\mathrm{CDS}},v_{\mathrm{Intron}},v_{\mathrm{RNA}},v_{\mathrm{Other}}\}$
Step 2	若序列不含 CDS 部分,转到 Step 5
Step 3	对 CDS 的序列片段 $v_{\mathrm{CDS}}=\{v_{c,1},v_{c,2},\cdots\}$,表达为氨基酸符号序列 $a=\{a_{c,1},a_{c,2},\cdots\}$
Step 4	对氨基酸符号进行 Huffman 编码,形成二进制数据 b_{CDS}
Step 5	若序列不含 Intron、RNA 与 Other 部分,转到 Step 8
Step 6	对此三部分,搜索其中的近似重复片段
Step 7	根据匹配结果修正 DNA 序列,转变为直接精确重复模式记录附加信息并编码为二进制数据 b_{Intron}、b_{RNA}和 b_{Other}
Step 8	将 b_{CDS}、b_{Intron}、b_{RNA}和 b_{Other}分别使用 LZMA 算法进行压缩
Step 9	输出压缩算法结果 $c=\{c_{\mathrm{CDS}},c_{\mathrm{Intron}},c_{\mathrm{RNA}},c_{\mathrm{Other}}\}$
Step 10	若部分压缩输出 $c_i\in c$ 的体积大于原始数据 b_i,则使用 b_i 代替 c_i 作为输出结果

对用以记录重复片断状态的附加信息,可通过优化其数据表示进一步减少存储空间.在实验中发现,当设置搜索的最短重复片段长度不大于 10 个碱基时,获得的匹配结果间距不会超过 128 个符号.在存储中使用片断间距离表示匹配位置,可将存储空间减小为 7bit.此外,序列中最大重复片断单边长度不会大于 64 个符

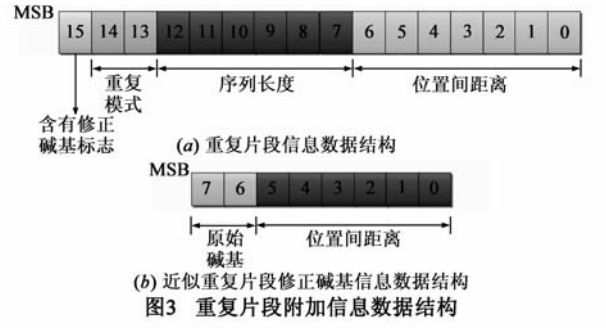


图3　重复片段附加信息数据结构

号,可存储于 6bit 空间中.加上代表四种重复模式的 2bit 数据,以及表示后续是否含有近似重复修正信息的 1bit 控制位,则每个重复片断的状态可存储于 2Byte 空间中.对于近似重复片断,需要另外存储碱基修正的信息.由于重复片断单边长度不大于 64 个符号,可用 6bit 空间存储修正碱基的位置,以及 2bit 空间表示原始的碱基.因此每个修正符号的信息可使用 1Byte 数据表示.在 BioLZMA 算法中,重复片断和修改碱基的附加信息数据结构如图 3 所示.

4 实验结果

实验中使用美国 GenBank 数据库[14]的 DNA 序列文件作为原始数据,其中包含了对序列的详细注释.在 BioLZMA 算法中,可直接读取这些注释,根据其生物学信息对序列进行压缩.

在实验中,将 BioLZMA 算法与典型 DNA 序列压缩方法:BioCompress-2、GenCompress、CTW + LZ、DNACompress 和 GeNML,以及原始的 LZMA 压缩算法分别作用于 11 个基准测试序列[15].使用压缩后序列中表示每碱基符号所需平均比特数(Bit per base,BPB),以及 BioLZMA 算法的压缩时间作为实验结果.基准测试序列包含了不同物种不同功能的 DNA 数据片断,能够有效评估压缩算法对含有不同数据特性的 DNA 序列的压缩能力.算法压缩率如表 3 所示.

由 BioLZMA 算法与原有 DNA 序列压缩算法的结果对比可以发现,新算法在大多数基准测试序列上的压缩率要好于原有方法.特别当序列包含生物信息学特征清晰时,算法压缩效果的提升更为明显.例如序列 VACCG 中生物学含义明晰的 CDS 集合占数据总量的 88%,因此算法在其上的压缩率明显优于其他方法.而对于含义划分不清,或是未包含注释信息的 DNA 序列数据,依赖于生物信息学特征的 BioLZMA 算法效果并不十分理想.另外,当 DNA 序列长度较大时算法性能提升更为明显.这是由于数据较长时,其包含的重复片断也较多,能够寻找到更多的隐含模式,从而可有效进行压缩编码.

表 3 压缩算法在基准测试序列上的压缩率(BPB)

测试序列	序列长度	Bio Compress-2	Gen Compress	CTW + LZ	DNA Compress	GeNML	LZMA	BioLZMA
CHMPXX	121024	1.6848	1.6730	1.6690	1.6716	1.6613	1.8650	1.5893
CHNTXX	155844	1.6172	1.6146	1.6120	1.6127	1.6102	1.9636	1.6035
HEHCMVCG	229354	1.8480	1.8470	1.8414	1.8492	1.8397	1.9820	1.6918
HUMDYSTROP	33770	1.9262	1.9231	1.9175	1.9116	1.9126	1.9681	1.9662
HUMGHCSA	66495	1.3074	1.0969	1.0972	1.0272	1.0124	1.6632	1.6520
HUMHPRTB	56737	1.9066	1.8466	1.8433	1.8165	1.7587	1.9186	1.8917
HUMHDABCD	58864	1.8770	1.8192	1.8218	1.7951	1.7132	1.8930	1.8367
HUMHBB	73308	1.8801	1.8204	1.8082	1.7897	1.7958	1.9082	1.8891
MPOMTCG	186608	1.9423	1.9105	1.9081	1.8876	1.8831	1.9794	1.8102
SCCHRIII	316613	1.9491	1.9487	1.9450	1.9485	1.9377	1.9608	1.7577
VACCG	191737	1.7837	1.7434	1.7389	1.7254	1.7603	1.8969	1.5216

由 BioLZMA 算法与原始的 LZMA 压缩算法的结果对比可以看出,在考虑了 DNA 序列的生物信息学特征,对序列数据进行了预处理后,产生的二进制数据流比原始序列数据更具有规律性,从而能够更为有效的进行压缩.BioLZMA 算法相比 LZMA 算法压缩率的提升,即是引入生物信息学特征的效果.

BioLZMA 算法在基准测试序列上的压缩时间如表 4 所示.实验运算平台为 CPU 主频 1.6GHz,内存为 1GB 的 PC 机.

由实验结果可以看出,BioLZMA 算法的压缩时间随序列长度的增加而有所增长,但总体而言运算时间维持在较低的水平.一般而言,若序列中含有的生物信息学特征较多,如具有不同生物学含义的片断越多,或重复序列越多,则需要更多的时间进行处理.但由此产生的计算时间增加并不明显.

根据 BioLZMA 算法在基准测试序列上的压缩性能结果可以发现,算法能在较短的时间内有效压缩 DNA 序列片断.对于序列较长,生物信息学特征明晰的序列,BioLZMA 算法的压缩时间会略有增加,但同时压缩率会有较大的提升.

表 4 BioLZMA 算法在基准测试序列上的压缩时间(Sec.)

测试序列	序列长度	压缩时间	测试序列	序列长度	压缩时间
CHMPXX	121024	2.07	HUMHDABCD	58864	0.78
CHNTXX	155844	2.71	HUMHBB	73308	0.82
HEHCMVCG	229354	3.82	MPOMTCG	186608	1.63
HUMDYSTROP	33770	0.69	SCCHRIII	316613	4.69
HUMGHCSA	66495	0.83	VACCG	191737	3.81
HUMHPRTB	56737	0.76			

5 总结

本文介绍了 DNA 序列数据的常见生物信息学特征.通过将这些特征引入 DNA 序列的预处理,提出了 BioLZMA 压缩算法.在算法中,含有不同生物学含义的片断被切分重组为 4 个集合,并分别进行压缩处理.其中编码区域的片断集合首先表达为氨基酸序列,而后通过 Huffman 编码为二进制数据.其他集合则通过预处理序列中的重复片断,将修正后的序列和附加信息编码为二进制数据.而后分别将各部分数据通过 LZMA 算法进行压缩.通过优化序列附加信息的表示方式,算法进一步提升了压缩率.实验表明,BioLZMA 算法能够在较短的时间内有效压缩 DNA 序列数据.与原有仅考虑 DNA 数据特点的压缩算法相比较,使用了生物信息学特征的 BioLZMA 算法压缩性能有所提升.特别是在生物信息学特征清晰的长序列上,其压缩结果优势更为明显.

参考文献

[1] Galperin M Y, Cochrane G R. Petabyte-scale innovations at the european nucleotide archive [J]. Nucleic Acids Research, 2009, 37: D1-D4.

[2] Srinivasa K G, Jagadish M, et al. Efficient compression of non-repetitive DNA sequences using dynamic programming [A]. Proc of International Conference on Advanced Computing and Communications [C]. Mangalore: ADCOM, 2006. 569 – 574.

[3] Grumbach S, Tahi F. Compression of DNA sequences [A]. Proc of Data Compression Conference [C]. Snowbird: DCC, 1993. 340 – 350.

[4] Chen X, Kwong S, et al. A compression algorithm for DNA sequences and its applications in genome comparison [A]. Proc of the 10th Workshop on Genome Informatics [C]. Tokyo: GIW, 1999. 51 – 61.

[5] Matsumoto T, Sadakane K, et al. Biological sequence compression algorithms [A]. Proc of Genome Informatics Workshop [C]. Tokyo: CIW, 2000. 43 – 52.

[6] Chen X, Li M, et al. DNACompress: Fast and effective DNA sequence compression [J]. Bioinformatics, 2002, 18 (12): 1696 – 1698.

[7] Korodi G, Tabus I. An efficient normalized maximum likelihood algorithm for DNA sequence compression [J]. ACM Transactions on Information Systems, 2005, 23 (1): 3 – 34.

[8] 林毅申,林丕源,等.基于字典的 DNA 序列压缩算法研究及应用 [J].计算机应用研究,2007,24 (6):265 – 267.

[9] Baxevanis A D, Ouellette B F F. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, Third Edition [M]. United States: Wiley Publishing House, 2005.

[10] 王玉,饶妮妮,等.基于小波变换技术预测 DNA 序列的编码区 [J].电子学报,2007,35 (1):141 – 144.
WANG Yu, RAO Ni-ni, et al. Predicting protein coding regions of DNA sequences based on wavelet translation technique[J]. Acta Electronica Sinica, 2007, 35(1): 141 – 144.

[11] Kaessmann H, Ebersberger I, et al. DNA sequence variation among humans and apes [A]. Proc of the Annual International Conference on Computational Molecular Biology [C]. Tokyo: RECOMB, 2002. 175.

[12] Micklos D, Freyer G A. DNA Science: Afirst Course, Second Edition [M]. United States: Cold Spring Harbor Laboratory Press, 2003.

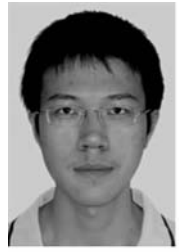
[13] Lempel-Ziv-Markov chain algorithm [DB/OL]. http://en.wikipedia.org/wiki/Lempel-Ziv-Markov_chain_algorithm, 1998.

[14] Benson D A, Karsch-Mizrachi I, et al. GenBank [J]. Nucleic Acids Research, 2008, 36: D25 – D30.

[15] Osborne M. Predicting DNA sequences using a backoff language model [DB/OL]. http://www.cogsci.ed.ac.uk/~osborne/dna-backoff.ps.gz, 2009-05-15.

作者简介

纪　震　男,1973 年 8 月出生江苏省溧阳市,工学博士,2004 年晋升教授,现为深圳大学博士生导师.主要研究方向:智能计算、图像处理、数字水印以及数字信号处理硬件系统.
E-mail: jizhen@szu.edu.cn



周家锐　男,1984 年 7 月出生于广东省韶关市,现为浙江大学博士研究生.主要研究方向:生物智能算法、DNA 数据压缩.
E-mail: 030bug@163.com