基于码书索引变换的高通量DNA序列数据压缩算法

谭 丽,孙季丰
(华南理工大学电子与信息学院,广东广州 510641)

摘 要: 提出一种高通量DNA序列数据的压缩算法.该算法先采用码书索引变换模型,将传统码书索引值的表示方法变换成由四个标准碱基字符替代的四进制数值方式,并采用一种界定替换串与非替换串的简明编码方法,接着通过信息熵的大小来决定是否进行块排序压缩变换(BWT),最后进行前移编码变换和Huffman熵编码.在多种测序数据集上的实验结果表明,CITD在大多数情况下可以获得比本文所对比的高通量DNA专用压缩方法更优的压缩性能.

关键词: 高通量DNA序列;码书索引变换模型;块排序压缩变换;前移编码;信息熵;数据压缩算法
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2015)05-1007-07
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.05.026

High-Throughput DNA Sequence Data Compression Method Based on Codebook Index Transformation

TAN Li, SUN Ji-feng
(*School of Electronic and Information Engineering, South China University of Technology, Guangzhou, Guangdong* 510641, *China*)

Abstract: A novel high-throughput DNA sequence compression method based on codebook index transformation (CITD) is proposed. In CITD, we used the codebook index transformation (CIT) model, to substitute the traditional represatation of codebook indexes by the quaternary values which are expressed by the four standard base characters, and adopted a simple encoding method to distinguish the replaced and non-replaced substring, and subsequently determined whether need to use the Burrow Wheeler Transformation (BWT) according to the value of information entropy, finally used move to front (MTF) transformation and Huffman entropy coding to compress the data. Experimental results on several sequencing data sets demonstrate better performance of CITD than the high-throughput DNA sequence compression algorithms cited in this paper, in most cases.

Key words: high-throughput DNA sequence; codebook index transformation (CIT) model; burrow wheeler transfarmation (BWT); move to front (MTF); information entropy; data compression algorithm

1 引言

目前实施的千人基因组计划、国际单体型图计划、孟德尔遗传疾病等项目,利用"下一代测序"技术产生了海量DNA测序数据.如何存储和传输高通量DNA测序产生的数据,成为决定DNA研究发展的重要因素之一.数据压缩是有效解决DNA序列存储和传输的一种重要方法[1,2].

由于DNA序列数据的特殊性,使用传统的压缩算法并不是很理想.因此从1993年起出现了专门针对DNA序列的压缩算法,如GeNML[3]、POMA[4]、BioLZMA-2[5]等.这些压缩算法只利用DNA序列自身的冗余信息(如直接重复、镜像、反转和互补回文),对小规模DNA测序数据的压缩效果较好,但高通量DNA序列数据往往是对全基因组的大规模测序,使用上述方法后的压缩效率有限,需要有专门针对高通量DNA序列数据的压缩方法.Kuruppu[6]等人提出基于优化的Lempel-Ziv算法的大数据DNA压缩方法(Optimized Relative Lempel-Ziv, RLZ-opt),该算法在基因组序列中利用扩展的自索引去控制搜索的子串,从而达到较好的压缩效果.2011年,文献[7]提出基于参考基因组序列的高通量DNA序列数据压缩的思想,利用提取参考基因组序列和重测序序列间的差异进行压缩,同时设计了相应的算法并开发了压缩工具GRS (Genome ReSequencing).该算法在不依赖其它外部数据的前提下,将原始2986.8MB大小的第一个韩国人个体基因组序列压缩至18.8MB.同样采用基于参考基因组的压缩思想,Jones[8]等人提出基于拼接的从头测序的高通量DNA序列压缩算法Quip.虽然这些

收稿日期:2014-01-07;修回日期:2014-04-17;责任编辑:蓝红杰
基金项目:国家自然科学基金青年科学基金(No.61202292);广东省自然科学基金(No.9151064101000037)

基于参考基因组的压缩方法得到的压缩比很高,但对于参考序列依赖性太强,而实际上有些测序数据并不存在现成的参考基因组;另外,由于压缩和解压缩都需要相同的参考基因组,参考基因组必须事先保存在本地,所以参考基因组确实将直接影响压缩数据的使用.接着,在 2012 年 Kuruppu[9]等人首次提出基于迭代字典结构的快速高通量 DNA 序列数据压缩算法,将基因组中出现次数较多的重复子串通过字典的多次迭代,用较少的字符取代这些重复子串,从而达到较好的压缩率.但该算法对序列中重复子串较少或重复子串长度较短时的压缩效果较差,并且对子串的频率分布属性利用不充分.

针对以上算法的不足,本文提出了另外一种基于码书索引变换的高通量 DNA 序列数据压缩算法(CITD).在 CITD 中,用四个标准碱基字符表示的四进制数值替代传统码书的索引值表示方法,同时采用一种界定替换串与非替换串的简明编码方法,极大限度地避免了编码后字符集的增长,用信息熵的大小来决定是否进行块排序压缩变换(BWT)[10],最终用前移编码(MTF)[11]变换和 Huffman 熵编码进行压缩.

2 码书索引变换模型

本文提出一种码书索引变换(CIT)模型.针对 FASTA 格式的 DNA 序列,其碱基字符集记为 $SET\_ALL = \{A,C,G,T,M,R,W,S,Y,K,V,H,D,B,N\}$.将标准碱基字符集和非标准碱基字符集分别记为 $SET\_BASE = \{A,C,G,T\}$、$SET\_NONBASE = \{M,R,W,S,Y,K,V,H,D,B,N\}$,并考虑压缩的序列对象主要包括 FASTA 格式的 DNA 序列[12],压缩除去以符号“>”开头的注释行剩余的数据.

本文对 CIT 模型阐释时采用的符号约定如下:

①码书中各模式串的串长度记为 L;②码书的大小记为 S(即模式串总数);③原始序列中的子串对应于 CIT 中的模式串,其匹配方式以字符 type 来表示,type 所属的字符集记为 SET_TYPE,取 $SET\_TYPE = SET\_BASE$,SET_TYPE 中的字符依次对应于 DNA 序列中的直接重复、镜像重复、互补重复和互补回文 4 种匹配方式.以标准碱基字符来描述匹配模式,避免了编码字符集大小的增长;④CIT 模式串的编码记为 words.考虑到 SET_BASE 中含有 4 个标准碱基字符,同时要抑制替换编码字符集的扩大,故我们采用四进制字符集 SET_BASE 表示 words.若采用大于四的进制,则需要引入额外替代字符,增加编码开销;若采用过小的进制,则 words 可能过长,影响压缩性能.其含义是用模式串在码书中的位置索引替换掉原始序列中相应的匹配串,索引值的进制与 words 的对应关系示例如表 1 所示.

表 1 CIT 模型中的编码 words

十进制	四进制	words
0	0	A
1	1	C
2	2	G
3	3	T
4	10	CA
5	11	CC
6	12	CG
7	13	CT
……	……	……

为加快 CIT 建表的速度,不依次扫描记录待压缩序列中长度为 L 的所有子串,而是将扫描的步进长度设定为 L,且只存储 SET_BASE 中的字符,包含 $SET\_NONBASE$ 中字符的子串跳过,不进入码书.在扫描记录的过程中,统计序列中出现重复(符合 4 种匹配模式之一的均作为一种模式串进入码书,且码书中只记录最早出现的模式串,这样可以有效地缩小码书大小)模式串的频率 freq 值,CIT 中的模式串以 freq 从大到小排序,使得重复概率高的模式串编码(words)长度较短.举例说明 CIT 模型建立的具体过程,如图 1 所示.

图 1 为 $L=6$ 的 CIT 模型建立的示例.以 $L=6$ 的步长对图 1 中的“>”序列扫描,记录模式串的索引和出现的频率 freq 值.模式串“TTGCAC”分别与图 1“>”序列中的模式串“CACGTT”、“TTGCAC”和“GTGCAA”构成“镜像重复”、“直接重复”、“互补回文重复”,故模式串“TTGCAC”在原始序列中重复出现 4 次,即在 CIT 中模式串“TTGCAC”的 freq = 4.同理,模式串“TCCTGA”和“GATAGT”在 CIT 中的 freq 值都为 2.图 1 中 CIT 表的第四列表示对 CIT 模式串进行编码,实际上是对模式串在 CIT 对应的索引号进行编码.

3 基于码书索引变换的高通量 DNA 序列数据压缩算法

本文利用 CIT 模型对高通量 DNA 序列数据进行压缩,来减少 DNA 序列中的冗余度.算法整体流程如图 2 所示.

图 2 中的 Seq_1 表示原始高通量 DNA 序列经过 CIT 模型编码后的输出序列;Seq_2 表示 Seq_1 经过 BWT 变换

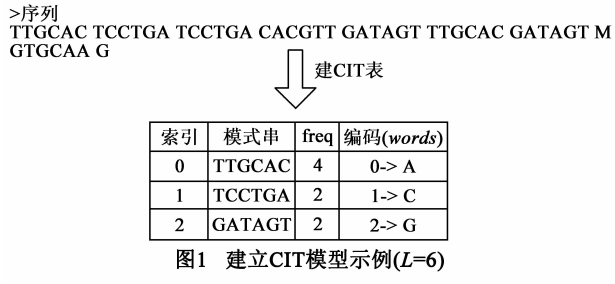

图1 建立CIT模型示例(L=6)

后得到的序列;$Seq_3^{'}$表示对编码序列 Seq_2进行 MTF 变换后的序列;Seq_3表示序列 Seq_1通过 BWT、MTF 变换后的结果;Seq_{out}代表 DNA 序列经过 CITD 算法压缩后的输出序列.

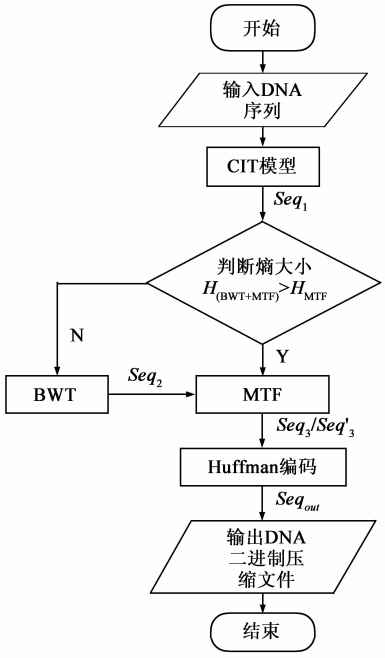

图2　CITD压缩流程图

3.1　CIT 模型的编码与解码

对 CIT 模型的设计一般需要依据以下思路:

①非替换子串与替换子串需要明确区分规划.

②尽量不采用或少采用新的替代字符,此处我们加入字符'0'作为非替换串的标志,字符'1'来标识两个重复碱基.

③替代操作的编码字符数尽量少.

3.1.1　CIT 模型的编码

为利于后续的 BWT 等变换,CIT 模型的编码分为段编码和连接编码.在段编码中,CIT 表中的模式串采用(type,words)的格式,非模式串直接输出;在连接编码中,编码原则:

①属于 *SET_NONBASE* 中的字符不进行替代,可作为解码时段的分界符之一.

②非替换串,在其前加数字前缀"0".

③为便于识别出连续的多个替换串中的各独立段,若当前替换串的首字符(*this_head*)与上一个替换串的尾字符(*last_tail*)不同,则在 *this_head* 前添上与 *last_tail* 相同的字符,凑成连续两个重复的碱基类型(如 AA、CC、GG、TT).为排除解码的二义性,替换串内部出现的 AA、CC、GG、TT,分别用 1A、1C、1G、1T 替代.

④在③的条件下,各段的段末字符不可能为单个字符 0 或 1.根据图 1 中得到的 CIT 表,对其编码,结果如图 3 所示.

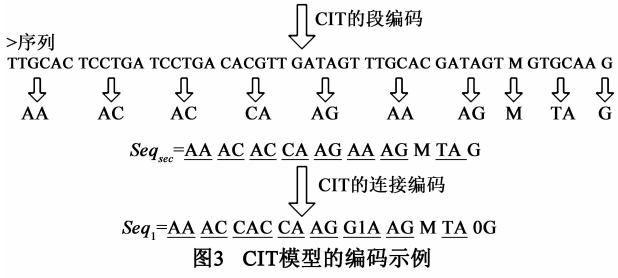

图3　CIT模型的编码示例

图 3 显示对图 1 的原始"＞"序列进行 CIT 段编码和连接编码后的结果,分别记为 Seq_{sec}、Seq_1.模式串"TTGCAC"在 CIT 表中出现 4 次,按规定编码为'A'字符且匹配方式为直接重复,故在段编码中编码为"AA",该模式串对应的镜像重复模式串"CACGTT"段编码为"CA";对应的互补回文重复模式串"GTGCAA"段编码为"TA",同理其它模式串也进行类似的段编码.对于 CIT 中的非模式串,直接编码即可.为方便后续解压,对 CIT 段编码后再进行二次编码,即连接编码.在图 3 段编码序列 Seq_{sec} = AA AC AC CA AG AA AG M TA G中,按照预先定义的连接编码规则,编码后结果为 Seq_1 = AA AC CAC CA AG G1A AG M TA 0G.

3.1.2　CIT 模型的解码

CIT 的解码涉及到两方面:

(1)段识别:①遇到连续两个相同的碱基字符(*SET_BASE* 中的字符)时,从中间一分为二,分别成为"前段"和"后段",由于不存在长度为 1 的段,所以划分后若前段长度为 1,则表明后段的首字符需并入前段处理;②遇到第一个 *SET_NONBASE* 中字符,将该字符左边划分为"前段";③记 *SET_NONBASE* 字符后面遇到第一个非 *SET_NONBASE* 字符为 x,若 $x \in SET\_BASE$,则 x 引起的一段为替换串,此时的解码规则同①;若 x = 0,表明该段为非替换串.

(2)遇到 1A、1C、1G、1T,则依次解码为 AA、CC、GG、TT.

3.1.3　CIT 模型分析

设 CIT 模型中第 i 个模式串的 words 为 θ_i,其长度为 λ_i,编码过程中被替换的次数为 η_i,含 *SET_NONBASE* 的段数及总长分别为 α 和 β,未替换的 *SET_BASE* 中字符串的段数及总长分别为 ε 和 ζ,字符集大小为 Φ_1,则原始串长度 len_0如式(1)所示:

$$len_0 = L \times \sum_{i=0}^{S-1} \eta_i + \zeta + \beta \tag{1}$$

CIT 模型段编码后的长度 len_1为:

$$len_1 = \beta + \zeta + \sum_{i=0}^{S-1} (\lambda_i \times \eta_i) \tag{2}$$

令 CIT 段编码后 *this_head* 与 *last_tail* 相同的概率为

P_1,则连接编码后字符集大小和总长度分别如式(3)和(4):

$$\phi_2 = \phi_1 + 2 \tag{3}$$

$$len_2 = (len_1 + \varepsilon) + (1 - P_1) \times \left(\sum_{i=0}^{S-1} \eta_i - \alpha \right) \tag{4}$$

模式串长度为 L 时,CIT 大小的理论最大值 $S_{\max}$和 λ_i的理论最大值如式(5)、(6).

$$S_{\max} = \lfloor 4^{L-1}/2 \rfloor + 4^{L/2} \tag{5}$$

$$(\lambda_i)_{\max} = \log_4 S_{\max} \tag{6}$$

其中符号⌊⌋表示向下取整.

从式(5)和(6)可知,λ_i的理论最大值是小于 L 值的,从而可使得原始序列经 CIT 模型编码后的输出较大幅度地缩短,且极大程度的避免了字符集的扩大,为后续的处理提供便利.假设待压缩原始序列中包含的标准碱基序列长度为 γ,则实际生成的 CIT 模型中的模式串总数 $S' < (\gamma/L)$.若 $\gamma/L > S_{\max}$,则 ε 和 ζ 的值均等于或趋近于 0,这对于 CIT 的连接编码是有利的,减少了 len_2的增长.因此,可以根据 γ 的值来估算选择较合适的 L 值,使得 CIT 可以覆盖到含有标准碱基的所有片段.由 $\gamma/L > S_{\max}$可知,只需 $L \times S_{\max} < \gamma$,即可达到上述目的,实验测得 $L \times S_{\max}$的取值,在区间($\gamma/20$,$\gamma/10$)时压缩效果较好.

考虑到 CIT 建表是以步长 L 来进行的,且索引编码以 freq 由大到小排序,故建表时间复杂度为 $O(len_0/L + S\log S)$,而段编码及连接编码的时间复杂度为 $O(len_0/L + \alpha + \varepsilon)$,由式(5)可知 S 的取值处于较小范围,除 S 外的其他影响因子均为线性影响,可见 CIT 的预处理编码耗时很少.CIT 解码时,由连接编码解为段编码耗时为线性时间复杂度,而段编码恢复为原编码的时间复杂度约为 $O(\alpha + \varepsilon)$,耗时很短.

3.2 BWT 与 MTF

BWT(Burrow Wheeler Transformation)[10]是对字符串轮转后得到的字符矩阵进行排序和变换,变换之后的字符串用通用的统计压缩模型(如 Huffman 编码)等进行压缩就能得到更好的压缩比.假设待压缩的串和转换后的串分别记为 Q、Z,并规定 Q 含有 k 个字符,则 BWT 变换步骤如下:

Step 1:将串 Q 中的字符依次向左循环一位,产生 $k \times k$ 矩阵,记为 P.

Step 2:对 $\boldsymbol{P}$ 中的每行按字典顺序进行排序(如 A,C,G,T),排序后得到新的矩阵,记为 $\boldsymbol{P}'$.

Step 3:保存 $\boldsymbol{P}'$的最后一列(即转换后的串 Z)和串 Q 的第一个字符所在的位置(记为 position).

对图 3 中 CIT 模型编码后的序列 Seq_1 = AA AC CAC CA AG G1A AG M TA 0G进行 BWT 变换,得到变换后的序列 Seq_2 = AGTGAC1CAAACCAAG0AAGM 和 position = 3.

MTF(Move To Front)[11]是前移编码的简称,其主要思想是将字符编码为在“最近使用字符”列表中的索引,索引为该字符在列表中对应的位置.这个“最近使用字符”列表在每次编码后会更新,被编码用过的符号前移至列表首位.例如对 BWT 变换后的序列 Seq_2 进行 MTF 变换,首先建立索引表如表 2 所示;接着进行编码,输出 MTF 的结果 Seq_3 = 012213443003030151056.对 Seq_1不进行 BWT 变换,直接 MTF 变换的结果 Seq_3' = 000101101020320345364.考虑到对于等长序列,其信息熵越小,则可压缩的空间越大,故比较序列 Seq_3和 Seq_3'的信息熵大小,得到 $H_{Seq_3} = 2.61\,bit$, $H_{Seq'_3} = 2.45$bit,即 $H_{Seq_3} > H_{Seq'_3}$,故在该示例中对 Seq_1序列只需进行 MTF 变换.信息熵的计算[13]如式(7):

$$H_m = -\sum_{i=1}^{n} p_{i,m}(a_i) \log_2 p_{i,m}(a_i) \tag{7}$$

其中 $m = 1,2$, $P_{i,m}$表示序列 m 中符号 a_i的概率,n 为 m 序列中符号种类的总个数.

表 2 序列 Seq_2的 MTF 索引表

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

经过 BWT 和 MTF 的变换,为接下来的 Huffman 编码创造了良好的基础.而且 BWT 和 MTF 变换步骤简单可逆,也降低了解压算法的复杂度.

3.3 Huffman 编码

Huffman 编码是一个 two-pass 的过程,首先统计数据中每个符号的概率,接着通过每个符号的概率,得到一棵 Huffman 树,该 Huffman 树保存每个符号的 Huffman 编码.通过 Huffman 构造数据的每个字符的 Huffman 编码,对于数据中出现次数较多的符号,其对应的 Huffman 编码位数较少.最后再扫描一遍数据,通过构造的 Huffman 树对数据中的每个符号进行替换,得到整个数据的 Huffman 编码.譬如对 MTF 变换后的序列 Seq_3' = 000101101020320345364,采用 Huffman 编码,其结果如表 3 所示.

表 3 序列 Seq'_3的 Huffman 编码结果

字符	0	1	2	3	4	5	6
频率	8	4	2	3	2	1	1
码字	0	111	100	101	1101	11001	11000

最终输出的二进制字符序列 Seq_{out} = 000111011111101110100010110001011101110011011100011101,所占用的字节数为 6.6 字节.

4　仿真实验及分析

测试机器是基于 Ubuntu 10.04 x86_64 的操作系统,拥有 4GB 的内存和主频为 2.8GHz 的双核 CPU.实验中首先将 CITD 与其它两种针对高通量 DNA 序列数据的压缩算法(RLZ-opt[6]和 COMRAD[9])作用于六种不同物种的测试基因组序列[14],以评估其压缩性能.另外,直接将 BWT、MTF 和 Huffman 编码算法(简记 BMH)应用于高通量 DNA 序列数据上,并与 CITD 算法进行对比.接着在其它三种测试基因组序列[7]上比较了 CITD 和 GRS[7]算法的性能.实验中测试基因组序列如表 4 所示.

在 CITD 算法中,我们设置 CIT 模型码书的各模式串长度 $L=10$.实验对比选用编码后数据集大小、压缩率、压缩和解压时间作为算法评价准则,其中压缩率(压缩后序列中表示每碱基符号所需平均比特数,Bit Per Base,BPB)是衡量算法压缩效率的指标之一,压缩率越小,说明算法压缩效果越好.压缩率 BPB 计算公式[15]如式(8):

$$\mathrm{BPB}=\frac{L_O}{L_I} \tag{8}$$

其中,L_I 为输入序列长度,L_O 为输出数据所占的比特数目.

四种算法(COMRAD、RLZ-opt、BMH 和 CITD)在六种测试基因组序列上的比较结果如表 5 所示,其中 Size 表示编码大小,Cmp.表示压缩率,*c-t* 和 *d-t* 分别表示压缩与解压时间,表中黑体表示最佳值.

由表 5 结果可发现,CITD 算法的压缩性能全部优于 COMRAD,主要是因为 COMRAD 算法中码书生成的时间较长且对码书模式串的频率属性利用不充分;相对 RLZ-opt 算法,CITD 在六个测试数据集上都能达到较好的压缩效果,尤其在较小的高通量 DNA 序列数据(如 Hemoglobin、Mitochondria 等)上不仅能获得较低的压缩率,而且压缩和解压的时间也较少,虽然在较大的 S. cerevisiae 和 S. paradoxus 基因组数据集上与 RLZ-opt 算法的压缩率相当,但在压缩和解压时间上更优;BMH 算法计算步骤较少但未较好考虑序列数据间的高度冗余性,所以虽然在时间性能上略优于 CITD 算法,但在压缩率上的比较,CITD 算法具有明显优势.

表 4　九种测试基因组序列

数据集	大小(MB)	描述与数据来源
Hemoglobin	7.38	血红蛋白基因组 http://ww2.cs.mu.oz.au/~kuruppu/comrad/hemoglobin.fa.gz
Mitochondria	25.26	后生动物线粒体蛋白质基因组 ftp://ftp.ncbi.nlm.nih.gov/genomes/MITOCHONDRIA/
E.coli	43.58	大肠杆菌基因组 ftp://ftp.ensemblgenomes.org/pub/bacteria/release-5/fasta/
Influenza	112.64	禽流感病毒基因组(Sequence 6 from Patent WO9637624) ftp://ftp.ncbi.nih.gov/genomes/INFLUENZA/influenza.fna.gz
S.cerevisiae	529.76	啤酒酵母基因组 ftp://ftp.sanger.ac.uk/pub/users/dmc/yeast/latest/cere_assemblies.tgz
S.paradoxus	492.27	奇异酵母基因组 ftp://ftp.sanger.ac.uk/pub/users/dmc/yeast/latest/para_assemblies.tgz
TAIR	115.10	拟南芥基因组 ftp://ftp.arabidopsis.org/home/tair/Genes/TAIR9_genome_release/TAIR9_chr_all.fas
TIGR	361.00	水稻基因组 http://rapdb.dna.affrc.go.jp/download/irgsp1.html
Korean Genome	2986.80	第一个韩国人个体基因组(含有 24 条染色体,包含 1 至 22,X 和 Y 染色体)及线粒体(M) ftp://ftp.kobic.kr/pub/KOBIC-KoreanGenome/

表 5　四种算法在六种测试基因组序列上的压缩结果

测试集		Hemoglobin	Mitochondria	Influenza	E.coil	S.cerevisiae	S.paradoxus
COMRAD	*Size*(MB)	1.07	5.77	6.03	6.63	15.29	18.33
	Cmp.(BPB)	1.16	1.83	0.43	1.36	0.25	0.34
	c-t(s)	26	101	181	102	1237	1781
	d-t(s)	19	21	22	21	28	31
RLZ-opt	*Size*(MB)	1.05	5.13	5.92	6.01	**9.33**	**13.44**
	Cmp.(BPB)	1.14	1.63	0.42	1.23	**0.15**	**0.25**
	c-t(s)	13	41	45	68	412	377
	d-t(s)	10	13	17	16	12	10

表 5(续)

测试集		Hemoglobin	Mitochondria	Influenza	E. coil	S. cerevisiae	S. paradoxus
BMH	*Size*(MB)	2.12	10.13	20.6	9.25	45.35	31.91
	Cmp.(BPB)	2.20	3.75	1.46	1.89	0.74	0.59
	c-t(s)	**8**	**30**	**30**	**18**	**82**	**85**
	d-t(s)	**5**	**5**	**7**	**5**	**6**	**5**
CITD	*Size*(MB)	**1.01**	**3.91**	**5.11**	**5.32**	9.41	14.19
	Cmp.(BPB)	**1.09**	**1.24**	**0.36**	**1.09**	**0.15**	0.26
	c-t(s)	11	33	37	21	103	129
	d-t(s)	6	9	12	6	7	6

表 6 CITD 和 GRS 压缩对比结果

测试集	GRS				CITD			
	编码大小	压缩率(BPB)	压缩时间(s)	解压时间(s)	编码大小	压缩率(BPB)	压缩时间(s)	解压时间(s)
TAIR	**6.5KB**	**0.01**	87	42	11.6KB	**0.01**	**51**	**21**
TIGR	4.4MB	0.15	2119	566	**0.51MB**	**0.02**	**106**	**17**
Korean	**18.8MB**	**0.10**	**1972**	1628	28.57MB	0.15	2582	**155**

CITD 和 GRS 算法在另外三组测试基因组序列上的压缩对比结果如表 6,表中黑体表示最佳值.

表 6 显示 GRS 算法在两种酵母测试数据集(TAIR 和 TIGR)上压缩率相差很大,说明在个体基因组数据和参考基因组数据之间的差异比例不一样,从而导致 GRS 性能不稳定.虽然 GRS 在 Korean 基因组数据集上,具有 159 倍(原始文件大小 2986.8MB/压缩后文件大小 18.8MB)的压缩比,但解压耗时太长,同时对参考基因序列依赖性太强,而有些高通量 DNA 序列数据并不存在现有的参考基因组数据.以上综合分析表明,CITD 算法的压缩性能总体上优于 GRS.

另外,实验测出在 Mitochondria 和 S. paradoxus 两个基因组测试集上参数 L 与压缩率(BPB)之间的变化关系如图 4 所示.

图 4 结果显示当模式串长度 L 的取值在 10 左右时,算法对两组基因数据集的压缩效果较好,同时也显示出,L 的变化趋势对压缩效果的影响符合 3.1.3 节的讨论.

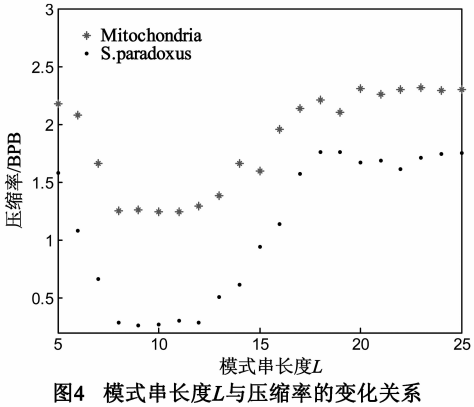

图4 模式串长度L与压缩率的变化关系

5 结论

本文提出一种基于码书索引变换的高通量 DNA 序列数据压缩算法 CITD.算法采用码书索引变换 CIT 模型,将传统码书索引值的表示方法变换成四个标准碱基字符替代的四进制数值方式,并采用一种界定替换串与非替换串的简明编码方法,极大限度地避免了编码后字符集的增长,用信息熵的大小来决定是否进行 BWT 处理,最终用 MTF 变换和 Huffman 熵编码进行压缩.进一步的工作是利用 CIT 模型,在基因组序列内部找出近似序列片段,利用片段间的相似性进行存储,使其提高压缩性能.

参考文献

[1] 朱泽轩,张永朋,等.高通量 DNA 测序数据压缩研究进展[J].深圳大学学报理工版,2013,30(4):409 – 415.
Zhu Ze-xuan, Zhang Yong-peng, et al. Advance in the compression of high-throughput DNA sequencing data[J]. Journal of Shenzhen University Science and Engineering, 2013, 30(4): 409 – 415.(in Chinese)

[2] 纪震,周家锐,等.DNA 序列数据压缩技术综述[J].电子学报,2010,38(5):1113 – 1121.
Ji Zhen, Zhou Jia-rui, et al. Overview of DNA sequence data compression techniques[J]. Acta Electronica Sinica, 2010, 38(5): 1113 – 1121.(in Chinese)

[3] 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.

[4] Zhu Zexuan, Zhou Jiarui, et al. DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(5): 643 – 658.

[5] 周家锐,纪震,等.基于 Memetic 优化的智能 DNA 序列数据压缩算法[J].电子学报,2013,41(3):513 – 518.
Zhou Jia-rui, Ji Zhen, et al. Intelligent DNA sequence data compression using memetic algorithm[J]. Acta Electronica Sinica,2013,41(3):513 – 518.(in Chinese)

[6] Kuruppu S, Puglisi S J, et al. Optimized relative Lempel-Ziv compression of genomes[A]. Proceeding of the 34th Australasian Computer Science Conference[C]. Australia: ACSC, 2011.91 – 98.

[7] Wang Congmao, Zhang Dabing. A novel compression tool for efficient storage of genome resequencing data[J]. Nucleic Acids Research,2011,39(7):E45 – U74.

[8] Jones D, Ruzzo W, et al. Compression of next-generation sequencing reads aided by highly efficient de novo assembly[J]. Nucleic Acids Research,2012,40(22):E171.

[9] Kuruppu S, Beresford-Smith B, et al. Iterative dictionary construction for compression of large DNA data sets[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2012,9(1):137 – 149.

[10] Li Cong, Ji Zhenzhou, et al. Efficient parallel design for BWT-based DNA sequences data multicompression algorithm[A]. Proceeding of International Conference on Automatic Control and Artificial Intelligence[C]. Xiamen: ACAI, 2012. 967 – 970.

[11] Wikipedia. Move-to-front Transformation[DB/OL]. http://en.wikipedia.org/wiki/Move-to-front-transform, 2013-12-09.

[12] Wikipedia. FASTA[DB/OL]. Http://blast.ncbi.nlm.nih.gov/blastcgihelp.shtml,2013-12-09.

[13] Shamir. Gil I. Universal lossless compression with unknown alphabets-the average case[J]. IEEE Transactions on Information Theory,2006,52(11):4915 – 4944.

[14] Kuruppu S, Beresford-Smith B, et al. Iterative dictionary construction for compression of large DNA datasets: supplementary material[OL]. http://www.computer.org/csdl/trans/tb/2012/01/ttb2012010137Abs.html,2013-12-09.

[15] 纪震,周家锐,等.基于生物信息学特征的 DNA 序列数据压缩算法[J].电子学报,2011,38(4):991 – 995.
Ji Zhen, Zhou Jia-rui, et al. Bioinformatics features based DNA sequence data compression algorithm[J]. Acta Electronica Sinica,2011,38(4):991 – 995.(in Chinese)

作者简介

谭　丽　女,1984 年生于湖南常德.现为华南理工大学信息与通信工程专业博士研究生.研究方向为生物信息学,计算智能,图像与视频处理.
E-mail:t.li07@mail.scut.edu.cn

孙季丰　男,1962 年生于广东揭阳,现为华南理工大学电信学院教授,博士生导师.研究方向包括智能信号处理、图像与视频处理、自组织通信网等.