

经典 Ramsey 数 DNA 计算模型(Ⅱ): 基于位序列的 DNA 计算模型

许 进^{1),2)} 范月科²⁾

¹⁾(北京大学信息科学技术学院高可信软件技术教育部重点实验室 北京 100871)

²⁾(华中科技大学分子生物计算机研究所 武汉 430074)

摘 要 Ramsey 数问题是组合数学乃至整个数学中最具魅力的研究领域,也是最困难的数学问题之一. 对于经典 Ramsey 数,至今只有 9 个 Ramsey 数得到解决. 按照传统的算法,其搜索空间太大,当前的电子计算机无法胜任. 研究表明,DNA 计算在求解困难的 NP-完全问题上优于电子计算机. 目前已经建立了众多求解 NP-完全问题的 DNA 计算模型,但未见用于求解 Ramsey 数的 DNA 计算模型. 作者建立了一种新颖的 DNA 计算模型,用于一般经典 Ramsey 数的求解. 全文共分两篇,该文属第二篇,在首篇工作的基础上,建立了所谓的经典 Ramsey 数位序列 DNA 计算模型,文中对模型的存储库的建立、解的检测子系统以及运算子系统等问题展开了较为详细地讨论,并给出了使用该模型求解经典 Ramsey 数详细的方法与步骤.

关键词 经典 Ramsey 数;DNA 计算;位序列 DNA 计算模型

中图法分类号 TP301

Classical Ramsey Number DNA Computing Model (Ⅱ): Add-Bit-Sequence DNA computing Model

XU Jin^{1),2)} FAN Yue-Ke²⁾

¹⁾(*Institute of Software, School of Electronics Engineering and Computer Science,*

Key Laboratory of High Confidence Software Technologies of Ministry of Education, Peking University, Beijing 100871)

²⁾(*Institute of Biomolecular Computer, Huazhong University of Science and Technology, Wuhan 430074*)

Abstract Classical Ramsey number problem is a NP complete problem. It takes exponential time to solve classical Ramsey number problem with traditional electronics computer. It is necessary to study new computation methods because traditional electronics computer faces with greatly difficulty in solving NP complete problem. DNA computing possesses high parallelism in data and higher storage capacity than normal systems. Hence, in theory, it is feasible to solve NP complete problems with DNA computing. A novel DNA computing model based on add-bit-sequence is proposed for the classical Ramsey number problems in this paper. The model consists of memory sub-system, operation sub-system and detection sub-system. The DNA computing model is a novel method for solving classical Ramsey number problems. The designed method and process of DNA sequence in the memory sub-system are provided, such as restriction, algorithm and probe. The operation sub-system is set up with PCR technique, and the sub-system for detect is described. As for the encoding problem of storage sub-system, the authors give three steps as follows: Give a number for encoding according to the possible Ramsey number's size, and estimate the length of each encoding string; Determine the restrictive conditions for DNA sequences in light of biological techniques, such as the types of hybridization; Ensure the length of each enco-

ding string on the basis of the above two conditions. As for how to delete the incorrect solutions, take the $r(5,5)$ for example; Delete the DNA strands which denote the K_5 from X ; after finishing the first step, add bit to the rest DNA strands. Additionally, take the same steps as the first step to delete the incorrect solutions K_5 or N_5 . In order to make the sub-system based on DNA computer work, the authors consider using the following biological techniques, such as restriction and PCR to delete the incorrect solutions.

Keywords Ramsey number; DNA computing; add-bit-sequence DNA computing model

1 引 言

众所周知,Ramsey 数^[1]的求解问题,是当今组合数学中最困难的问题之一,自从 1928 年至今,广义的 Ramsey 理论研究取得了一些成就^[2],但其困难程度使其发展并不理想. 人们在理论上求解 Ramsey 数问题遇到困难时,就自然地借助电子计算机^[3-5],但当顶点数增加引起子图数目快速增长时,电子计算机就力不从心了. 这也是目前 Ramsey 数问题研究停滞的主要原因之一^[6].

DNA 计算发展已有十多年^[7-12],随着分子生物学技术的发展,其研究前景越来越被看好. 它打破了传统意义上的计算机概念,为计算机科学的发展提供了新的思路. 到目前为止,已有不少报道利用 DNA 计算对图的最大团和最大独立集问题展开研究^[13-14],但尚无利用 DNA 计算对 Ramsey 数进行的研究. 本文在文献[6]研究的基础上,将 DNA 计算用于 Ramsey 数的研究,期望建立能够有效求解 Ramsey 数的 DNA 计算模型,为 Ramsey 数的研究提供新的方法和理论支持.

本文所提出的基于位序列 Ramsey 数 DNA 计算模型主要由 3 个部分构成:存储子系统、运算子系统和解的检测子系统,如图 1 所示. 文中对各个子系统进行理论论证,进而建立求解 Ramsey 数的 DNA 计算模型. 模型中存储子系统重点研究编码与 DNA 分子的合成. 由于 DNA 分子的合成可通过生物公司来处理,我们在这里只给出存储子系统中的一个关键问题,即所谓的基本型编码问题;运算子系统是模型中的一个关键问题,文章中通过采用可定量信息处理的 PCR 技术来实现运算;解的检测问题,文中主要采用常规的 PCR、测序技术等来完成.



图 1 加位型 Ramsey 数 DNA 计算模型的基本结构

2 存储子系统中编码问题的研究

十多年关于 DNA 计算的研究表明,编码问题是整个 DNA 计算研究中的核心问题,其主要原因如下:

- (1) 它直接影响着 DNA 序列的合成质量;
- (2) 编码的好坏直接影响着能否按照所设计的目标进行生物学反应;
- (3) 编码的好坏直接影响着解空间的大小. 希望通过编码来解决所谓的“解空间指数爆炸问题”(即随问题规模的增大 DNA 分子量成指数增长),此问题与 DNA 计算机能否深入发展息息相关;
- (4) DNA 计算的一个主要的难点是解的检测问题,而编码与酶切位点,或者发夹 DNA 分子,或者荧光标记等的结合有望较好地解决这个问题.

对于存储系统中的编码问题的研究一般按照下列步骤进行:

- (1) 根据 Ramsey 数的可能规模来确定编码的数目,并估算每个编码的长度;
- (2) 根据运算方式,即利用生物技术中杂交的类型来确定产生可用于生化反应的 DNA 序列的编码约束条件;
- (3) 在前两个条件基础上,确定编码的长度并给出相应的编码.

下面,我们来分步讨论.

编码的模式与数目

我们针对边序列 L_q 进行编码. 对文献[6]所确定的边序列,每个位 i 用两个 DNA 序列来表示,分别记作

$$r_i, b_i, i = 1, 2, \dots, q = C_p^2 \tag{1}$$

其中 r_i 表示在第 i 位存在边;而 b_i 则表示在第 i 位不存在边. 其实,这种表示实际上已经给出了图的相邻矩阵的编码.

众所周知,一个图 G 可唯一地用一个相邻矩阵 $A(G)$ 来表示. 对于一个简单无向图而言,由于 $A(G)$ 是对称的,且主对角线上的元素全为 0,故我们只需

要下三角阵(或上三角阵)即可,即若 $\mathbf{A}=\mathbf{A}(G)=(a_{ij})_{p\times p}$ 为

$$\mathbf{A}=\begin{pmatrix}a_{11}&a_{12}&a_{13}&\cdots&a_{1(p-1)}&a_{1p}\\a_{21}&a_{22}&a_{23}&\cdots&a_{2(p-1)}&a_{2p}\\a_{31}&a_{32}&a_{33}&\cdots&a_{3(p-1)}&a_{3p}\\\vdots&\vdots&\vdots&\cdots&\vdots&\vdots\\a_{p1}&a_{p2}&a_{p3}&\cdots&a_{p(p-1)}&a_{pp}\end{pmatrix},$$

则只需要其中的上三角阵中的数据即可：

$$\begin{matrix}a_{21}\\a_{31}&a_{32}\\\vdots&\vdots\\a_{p1}&a_{p2}&a_{p3}&\cdots&a_{p(p-1)}\end{matrix}$$

其中,

$$a_{ij}=\begin{cases}1, & \text{图的顶点 } v_i \text{ 与 } v_j \text{ 相邻,}\\0, & \text{否则}\end{cases},\quad i\neq j, i,j=1,2,\cdots,p\tag{2}$$

这些数据代表了图的所有可能边的位置信息. 这就是说,一个长度为 C_p^2 的 0-1 序列唯一地对应一个标定图, p 个顶点的标定图共有 $2^{C_p^2}$ 个. 显然,随着 p 的增大,这是一个很大的数字. 若要从这么多图集合中找出我们所需要的 Ramsey 图,即既不含 K_m ,又不含 N_n 的 Ramsey 数 $r(m,n)$ 的一个图,其搜索次数也是一个很大的数字,当今的电子计算机几乎无法涉入,而 DNA 计算机的优势却显露出来.

我们的编码设计方案如下所示：

$$\begin{matrix}1&2&3&\cdots&C_p^2\\r_1&r_2&r_3&\cdots&r_{C_p^2}\\b_1&b_2&b_3&\cdots&b_{C_p^2}\end{matrix}$$

显然,在本模型中,具有 p 个顶点的图需要的编码数为

$$2\times\frac{1}{2}p(p-1)=p(p-1)\tag{3}$$

例如,当 $p=40$ 时,需要的编码数为 1560 个. 若每个编码用长度为 60bp 来代表,则其长度为 $60\times 780=46800\text{bp}$.

下面,我们通过一个简单的例子给予说明,如图 2 所示.

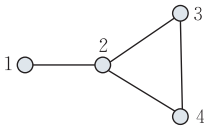


图 2 一个 4-阶图

由此图的相邻矩阵可知, $a_{12}=a_{23}=a_{24}=a_{34}=1$, $a_{13}=a_{14}=0$. 即表示该图的 0-1 序列是

$$l_1l_2l_3l_4l_5l_6=a_{12}a_{13}a_{23}a_{14}a_{24}a_{34}=101011\tag{4}$$

用上述 DNA 序列表示此图,即为 $r_1b_2r_3b_4r_5r_6$.

这里,我们仅给出 DNA 序列的形式表示,而略去具体的 DNA 序列的编码.

根据运算方式,即杂交的类型来确定 DNA 序列的约束条件

影响编码的主要因素有化学自由能、温度、生物酶、编码的距离、DNA 分子的组成等. 我们在本文中主要考虑降低分子杂交时的非特异性来进行编码,按照如下 3 个步骤进行:(1) 明确所编的码所受到的各种约束条件;(2) 在所确定的约束条件的基础上,将约束条件转化成相应的数学约束条件;(3) 在约束条件的基础上,给出编码的具体算法,并进行编码.

这里我们根据文献[15]的实验条件,建立利用该模型进行求解 Ramsey 数编码的约束条件如下：

- (1) 所设计的每个编码 x_i 都由 20 个碱基组成,其中 A、T、G、C 4 种碱基随机分布；
- (2) 所有编码没有超过 4 个 A,4 个 T,4 个 C 或 4 个 G 连续出现；
- (3) 任意编码的 GC 含量在 40%~60%之间；
- (4) 任意两条编码没有连续 8 个以上的碱基是相同的,也就是说,编码的最小子串为 8；
- (5) 任意一条编码自身不存在连续 4 个碱基的互补序列.

约束条件建立算法模型,以确定 DNA 序列

根据上述约束条件,我们在此以某图 G 的前 65 条边为例给予说明.

利用 Zhang 等人设计的程序^[16],用长度为 20 个碱基的寡核苷酸序列表示图 G 中前 65 条边的可能的着色,记为 $x_i,x\in\{r_i,b_i\}$, $i=1,2,\cdots,65$,它的 Watson-Crick 互补序列则用 \bar{x}_i 表示. 每个 x_i,\bar{x}_i 所对应的具体的寡核苷酸序列如表 1 所示.

表 1 DNA 序列

原链/上游	DNA 序列	补链/下游	DNA 序列
r1	5'-CTggTCCTCTCCTAATCC-3'	r1	
b1	5'-AAGAgAgAACGgAACTgTCC-3'	b1	5'- ggACAgTTCggTTCTCTCTT-3'
r2	5'-ACTTgAgCACTgACCTgACA-3'	r2	5'-TgTCAggTCAgTgCTCAAgT-3'
b2	5'-AAGAggCTACggACACTACT-3'	b2	5'-AgTAgTgTCCgTAgCCTCTT-3'
r3	5'-AAGgAgTAACCATCgCACAg-3'	r3	5'-CTgTgCgATggTTACTCCTT-3'
b3	5'-TAggTgCTACAgATTGtTCC-3'	b3	5'-ggACgAATCTgTAgCACCTA-3'

(续 表)

原链/上游	DNA 序列	补链/下游	DNA 序列
r4	5'-AAGTCTgAACgCCTACTCAC-3'	r4	5'-gTgAgTAggCgTTCAgACTT-3'
b4	5'-CAGAACACAggTATgCgATT-3'	b4	5'-AATCgCATACCTgTgTTCTg-3'
r5	5'-AAGACCACACCACgCATT-3'	r5	5'-gAATgCTgTggTgTggTCTT-3'
b5	5'-CgTgATTgTTggACTATTgg-3'	b5	5'-CCAATAgTCCAACAATCACg-3'
r6	5'-CCTTgTAGACCCAgATgTTC-3'	r6	5'-gAACATCTgggTCTACAAGg-3'
b6	5'-CgTTgCTCTgAATAgTTgCC-3'	b6	5'-ggCAACTATTCAgAgCAACg-3'
r7	5'-AATACgCACTCATCACATc-3'	r7	5'-CgATgTgATgAgTgCgTATT-3'
b7	5'-gACCTTACCgTTTAgAgTCg-3'	b7	5'-CgACTCTAAACggTAAggTC-3'
r8	5'-AATACATCAgAgCggAgACC-3'	r8	5'-ggTCTCCgCTCTgATgTATT-3'
b8	5'-ATggTggAAATCTACTcCC-3'	b8	5'-ggCgAgTAgATTTCCACCAT-3'
r9	5'-AAGgCTACAAACTCACcAC-3'	r9	5'-gTCggTgAgTTTgTAgCCTT-3'
b9	5'-ATgAggTTTgTTAgCCAgTC-3'	b9	5'-gCACTggCTAAACCTCAT-3'
r10	5'-ACAgAAAgAAACTCgCTTCg-3'	r10	5'-CgAAGCgAgTTTCTTTCTgT-3'
b10	5'-gAAGATgAACCAgCCTAACc-3'	b10	5'-ggTTAaggCTggTTCATCTTC-3'
r11	5'-AAGTgAACAgTgTgACCACC-3'	r11	5'-ggTggTCACACTgTTCACTT-3'
b11	5'-TCACATTAgTgTCACAgCgg-3'	b11	5'-CCgCTgTgACACTAATgTA-3'
r12	5'-CAGAgACAAGACgAACCTgT-3'	r12	5'-ACaggTTCgTCTTgTCTCTg-3'
b12	5'-TAGAAGAAgCAACCgTCTgT-3'	b12	5'-ACAgACggTTgCTTCTTCTA-3'
r13	5'-AACTTgTTCCACACACCCTC-3'	r13	5'-gAgggTgTgTggAACAAGTT-3'
b13	5'-gCTTATgTATCCTggCACTg-3'	b13	5'-CagTgCCaggATACATAAGC-3'
r14	5'-ATgAgTTACAAGCACCAcCg-3'	r14	5'-gCgTggTgCTTgTAACTCAT-3'
b14	5'-TACAgggTCTTCAGAACgAT-3'	b14	5'-ATCgTTCTgTAAGACCTgTA-3'
r15	5'-ATgTCTCgTCaggATgTCgT-3'	r15	5'-ACgACATCCTgACgAgACAT-3'
b15	5'-TTCCCTACTACCTTCCCAAg-3'	b15	5'-CTTgggAaggTAgTAgggAA-3'
r16	5'-ATgCCTCAACAACCTCCTgCT-3'	r16	5'-AgCaggAgTTgTTgAggCAT-3'
b16	5'-ATggTATgAAGCCTgACTCg-3'	b16	5'-CgAgTCaggCTTCATACCAT-3'
r17	5'-gAATACCgAgAgTgTTgTCC-3'	r17	5'-ggACAACACTCTCggTATTC-3'
b17	5'-ggATTgTATTggCgATgATg-3'	b17	5'-CATCATCgCCAATACAATCC-3'
r18	5'-TACATTCAaggACgACAggT-3'	r18	5'-ACCTgTgTCCTTgAATgTA-3'
b18	5'-CAAAGTgTAGgCAgggTAAC-3'	b18	5'-gTTACCTgCCTCAACTTTg-3'
r19	5'-AAGCggTAGACAgATTAC-3'	r19	5'-gTgAATCgTgTCTACCgCTT-3'
b19	5'-TAGAgTCCACCgAAGATAgC-3'	b19	5'-gCTATCTTCggTggACTCTA-3'
r20	5'-ACTgCTAATgACTCgTTCCg-3'	r20	5'-CggAACgAgTCATTAgCAgT-3'
b20	5'-AgTATCTgTCCTgTCTCACC-3'	b20	5'-ggTgAgACAggACAgATACT-3'
r21	5'-gTggTCgTAGATgTCACTCC-3'	r21	5'-ggAgTgACATCTACgACCAC-3'
b21	5'-AACTAACgACCgAgCCgAT-3'	b21	5'-ATCggCTCTggTCgTTAgTT-3'
r22	5'-TAGTCATAAgTgACCTcggC-3'	r22	5'-gCCgAggTCACTTATgACTA-3'
b22	5'-ATCTCCATCCAACCATCCA-3'	b22	5'-CTggATggTTggATggAgAT-3'
r23	5'-AATCAACTggTCACgACTgC-3'	r23	5'-gCgAgTCgTgACCAgTTgATT-3'
b23	5'-gTTATgAgTCgCgACACAg-3'	b23	5'-CgTgTgCTgCgACTCATAAC-3'
r24	5'-TAGTgCggAACCTATCTTgC-3'	r24	5'-gCAAgATAggTTCCgCACTA-3'
b24	5'-AAGTggAgACACTCACTACC-3'	b24	5'-ggTAGTgAgTgTCTCCACTT-3'
r25	5'-AAGTATCAgACAgCCATCCg-3'	r25	5'-CggATggCTgTCTgATACTT-3'
b25	5'-AAGTTgAAggCTTACgAgAC-3'	b25	5'-gTCTCgTAAgCCTTCAACTT-3'
r26	5'-ATgTCAACgATACCgTCACC-3'	r26	5'-ggTgACggTATCgTTgACAT-3'
b26	5'-ATgTgTATgTTgCgACAAGC-3'	b26	5'-gCTTgTCgCAACATACACAT-3'
r27	5'-gATTATCgTCCAgCCTTCTC-3'	r27	5'-gAgAAggCTggACgATAATC-3'
b27	5'-CCTCCgTAGTTATTgATGCC-3'	b27	5'-ggCATCAATAACTACgAgg-3'
r28	5'-gTTACggTTgACTCTgCTgA-3'	r28	5'-TCAgCAGAgTCAACCgTAAC-3'
b28	5'-CACTTCTACCCTCAACCTCA-3'	b28	5'-TgAggTTgAgggTAgAAGTg-3'
r29	5'-TAGTAgAAAgCCgACCACTC-3'	r29	5'-gAgTggTCggCTTTCTACTA-3'
b29	5'-CgTggAAGTCACTAAggTCT-3'	b29	5'-AgACCTTAgTgACTTCCACg-3'
r30	5'-ATTCTTCACAgAggTgCTgg-3'	r30	5'-CCAgCACCTCTgTgAAgAAT-3'
b30	5'-TAAgTgAgAATgCCAgTTgC-3'	b30	5'-gCAACTggCATTTCTCACTTA-3'
r31	5'-CgAgATgTTgTAAAggCTgC-3'	r31	5'-gCAgCCTTTACAACATCTCg-3'
b31	5'-ggTATgTAACAAGCgCACg-3'	b31	5'-CgTgCgTCTgTTACATACC-3'
r32	5'-ggTCATTATgggCATAgTgg-3'	r32	5'-CCACTATgCCCATAATgACC-3'
b32	5'-AACATTTACTCgTCgTTCgC-3'	b32	5'-gCgAACgACgAgTAAATgTT-3'
r33	5'-gCTCaggAgTgTTTATgACC-3'	r33	5'-ggTCATAAACACTCCTgAgC-3'
b33	5'-TAGTCCATCggCAAggTTCT-3'	b33	5'-AgAACCTTgCCgATggACTA-3'
r34	5'-AACACCACAgAgCATTCACg-3'	r34	5'-CgTgAATgCTCTgTggTgTT-3'
b34	5'-AAGTgCTgAAACCgTggAgT-3'	b34	5'-ACTCCACggTTTCAgCACTT-3'

(续 表)

原链/上游	DNA 序列	补链/下游	DNA 序列
r35	5'-TACTgAgAACgCTCgCTCTT-3'	r35	5'-AAGAgCgAgCgTTCTCAgTA-3'
b35	5'-ATgTgATAATgCCgTTCCTg-3'	b35	5'-CAGgAACggCATTATCACAT-3'
r36	5'-gATgTTgCCATAACTgCCTg-3'	r36	5'-CAGgCagTTATggCAACATC-3'
b36	5'-CTACTTATTCgTCAGCgTCg-3'	b36	5'-CgACgCTgACgAATAAgtAg-3'
r37	5'-gATgATTACACTCgCACagg-3'	r37	5'-CCTgTgCgAgTgTAATCATC-3'
b37	5'-gAACATAATggACCgACCTC-3'	b37	5'-gAggTCggTCCATTATgTTC-3'
r38	5'-ATCTgTgCTATCTCgTgCTC-3'	r38	5'-gAgCACgAgATAgCACAgAT-3'
b38	5'-AACTTACCATTggCTTCTgC-3'	b38	5'-gCAGAAgCCAATggTAAgTT-3'
r39	5'-gATAAACgAgTTCgCATAgC-3'	r39	5'-gCTATgCgAACTCgTTTATC-3'
b39	5'-gCAGTAGACgATACgACTCC-3'	b39	5'-ggAgTCgTATCgTCTACTgC-3'
r40	5'-CgAAgAAgATTACCCAgAgg-3'	r40	5'-CCTCTgggTAATCTTCTTCg-3'
b40	5'-AAGCACTCAACAgTACgAgC-3'	b40	5'-gCTCgTACTgTTgAgTgCTT-3'
r41	5'-ggTgATggTTgAAAgTCTCC-3'	r41	5'-ggAgACTTTCAACCATCACC-3'
b41	5'-TAGAgATTggACggAAgACg-3'	b41	5'-CgTCTTCCgTCCAATCTCTA-3'
r42	5'-TAGgTATAggTCgTTgAgCC-3'	r42	5'-ggCTCAACgACCTATACCTA-3'
b42	5'-CAAgTCACAATCgTAggTgC-3'	b42	5'-gCACCTACgATTgTgACTTg-3'
r43	5'-gCTAACAgTggTCAGACACg-3'	r43	5'-CgTgTCTgACCACTgTTAgC-3'
b43	5'-AATACCACCTgACTgCgTAG-3'	b43	5'-CTACgCagTCaggTggTATT-3'
r44	5'-AATACACTATCCAgCgACgg-3'	r44	5'-CCgTCgCTggATAgTgTATT-3'
b44	5'-ggTgTAAgCCTCCgTATTAg-3'	b44	5'-CTAATACggAgggCTTACACC-3'
r45	5'-ggAACCTACTCTggATgAAg-3'	r45	5'-CTTCATCCAAGTAgggTTCC-3'
b45	5'-ggTAACgATCCTgATAACgC-3'	b45	5'-gCgTTATCaggATCgTTACC-3'
r46	5'-gCTgTCCAACCAggTCTTAC-3'	r46	5'-gTAAgACCTggTTggACAgC-3'
b46	5'-CCTACACATCAATCAGCACC-3'	b46	5'-ggTgCTgATTgATgTgTAgg-3'
r47	5'-CCTTACAAATCgCCTATggT-3'	r47	5'-ACCATAggCgATTTgTAAgg-3'
b47	5'-CCAAACTTgCTTACTTCAgg-3'	b47	5'-CCTgAAgTAAgCAAgTTTgg-3'
r48	5'-CAAAGAgTTagTCgggTCTg-3'	r48	5'-CagACCCgACTAACTCTTg-3'
b48	5'-CTTCTATgTTTAgCCCgAgg-3'	b48	5'-CCTCgggCTAAACATAgAAg-3'
r49	5'-gCaggACAaggCTCATAgTT-3'	r49	5'-AACTATgAgCCTTgTCCTgC-3'
b49	5'-gTgACgCCATCATTTgAgAT-3'	b49	5'-ATCTCAAATgATggCgTCAC-3'
r50	5'-CTATCAGAAACCCgTCAGAg-3'	r50	5'-CTCTgACgggTTTCTgATAg-3'
b50	5'-gCTgACTTCACggATTTggA-3'	b50	5'-TCCAAATCCgTgAAgTCAGC-3'
r51	5'-CgAAggACTTAgTAACgAgg-3'	r51	5'-CCTCgTTACTAAgTCCTTCg-3'
b51	5'-CCTATgCCTAAATggTgTCg-3'	b51	5'-CgACACCATTTAggCATAgg-3'
r52	5'-CCTgTCCgATAgAATAgTgC-3'	r52	5'-gCACTATTCTATCggACagg-3'
b52	5'-AgTTgCgTCCACgAAAgTAg-3'	b52	5'-CTACTTTCgTggACgCAACT-3'
r53	5'-gCACTCCCAATgTgTTATgA-3'	r53	5'-TCATAACACATTgggAgTgC-3'
b53	5'-AggCTCCATCTTgAgAACTg-3'	b53	5'-CagTTCTCAAATgAggAgCCT-3'
r54	5'-gCTggCgACTACTATTTACg-3'	r54	5'-CgTAAATAgTagTCgCCAgC-3'
b54	5'-gCTCTCCCTTATggAATgAT-3'	b54	5'-ATCATTCATAAgggAgAgC-3'
r55	5'-CACTAAACAACgCagggTTC-3'	r55	5'-gAACCTgCgTTgTTTAgTg-3'
b55	5'-CAGCAACCACATCggTgATA-3'	b55	5'-TATCACCGATgTggTTgCTg-3'
r56	5'-gACCTCCTgAAAgAgTACgA-3'	r56	5'-TCgTACTCTTTCAggAggTC-3'
b56	5'-gTCACCTgCTAggAggATTC-3'	b56	5'-gAATCCTCCTAgCaggTgAC-3'
r57	5'-gAgTCgTCggAgATAaggTT-3'	r57	5'-AACCTTATCTCCgACgACTC-3'
b57	5'-gAATACCTgTgCTACCgAgT-3'	b57	5'-ACTCggTAGCACaggTATTC-3'
r58	5'-ggATAgCgATTgACTgAACg-3'	r58	5'-CgTTCAgTCAATgCgCTATCC-3'
b58	5'-CTgAgTCCTTTgAgTAAgCC-3'	b58	5'-ggCTTACTCAAaggACTCag-3'
r59	5'-CAGATAgACTCCgCTgAggT-3'	r59	5'-ACCTCagCggAgTCTATCTg-3'
b59	5'-gAgTTCCATTgTggCagAAg-3'	b59	5'-CTTCTgCCACAATggAACTC-3'
r60	5'-gCATTTACAgTCTTCTCgC-3'	r60	5'-gCgAgAAgACTgTgAAATgC-3'
b60	5'-CCAgACgATCTACgCTgATg-3'	b60	5'-CATCAGCgTAgATCgTCTgg-3'
r61	5'-CAGTTACATTgAgCggAAgC-3'	r61	5'-gCTTCCgCTCAATgTAACTg-3'
b61	5'-CAAgTATggCTCACATTCgT-3'	b61	5'-ACgAATgTgAgCCATACTTg-3'
r62	5'-CAAACAggCgTCTCTTTAg-3'	r62	5'-CATAAAgAgACgCCTgTTTg-3'
b62	5'-CAAgCAGCAGCgATgACTCTA-3'	b62	5'-TATAgTCACTCgTgCTgCTTg-3'
r63	5'-gACTTgCTCTgCgTgAgATT-3'	r63	5'-AATCTCACgCagAgCAAgTC-3'
b63	5'-gACATTgCTgAATCagTggT-3'	b63	5'-ACCACTgATTcagCAATgTC-3'
r64	5'-gCTACTgCTAagggTAATgC-3'	r64	5'-gCATTACCCTTAgCagTAGC-3'
b64	5'-gCACTgTATgACAggTCACg-3'	b64	5'-CgTgACCTgTCATACAgTgC-3'
r65	5'-CgTTAggACCTgggATAATC-3'	r65	5'-gATTATCCCAggTCCTAACg-3'
b65	5'-gATTACTCCACCCTCgTgTA-3'	b65	5'-TACACgAgggTggAgTAATC-3'

为了验证这些编码的可靠性,我们用生化实验计算机辅助软件 Primer Premier5.0 对可能构成引物对的寡核苷酸序列进行了测试,主要考虑的参数为 Hairpin, Dimer, 和 CrossDimer. 测试结果如图 3 所示.

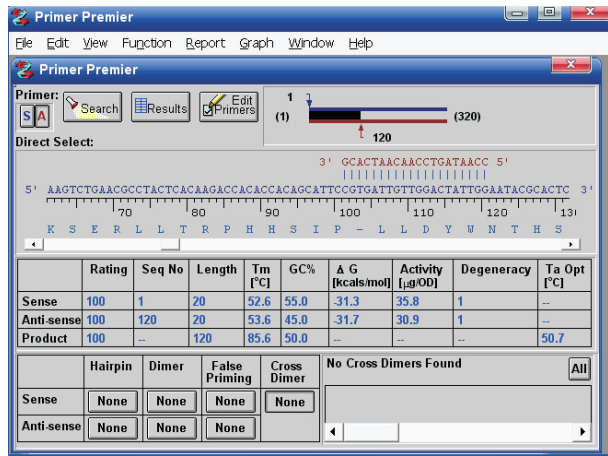


图 3 编码的测试结果

利用该生物学辅助软件,我们发现,部分编码质量较好,没有二级结构.但是有些引物对还存在二级结构的情况.我们根据前面所述的编码约束条件中的参数,对编码质量加以控制,选取具有二级结构但 $|\Delta G|$ 较小的编码用于后续的实验.

由确定的 DNA 序列以及杂交的类型来确定探针

为了构建初始解空间,我们根据所得到的编码,按照如下的方法确定探针:按照图的标定顺序,用后一个 DNA 序列的后 10 个碱基与前一个 DNA 序列的前 10 个碱基组合成长度为 20bp 的 DNA 序列,然后对每个序列反向取补.其中,所需要的探针分别如下:

$\overline{r_1 r_2}, \overline{r_1 b_2}, \overline{b_1 r_2}, \overline{b_1 b_2}; \overline{r_2 r_3}, \overline{r_2 b_3}, \overline{b_2 r_3}, \overline{b_2 b_3}; \overline{r_3 r_4},$
 $\overline{r_3 b_4}, \overline{b_3 r_4}, \overline{b_3 b_4}; \overline{r_4 r_5}, \overline{r_4 b_5}, \overline{b_4 r_5}, \overline{b_4 b_5}; \overline{r_5 r_6}, \overline{r_5 b_6}, \overline{b_5 r_6},$
 $\overline{b_5 b_6}; \overline{r_6 r_7}, \overline{r_6 b_7}, \overline{b_6 r_7}, \overline{b_6 b_7}; \overline{r_7 r_8}, \overline{r_7 b_8}, \overline{b_7 r_8}, \overline{b_7 b_8}; \overline{r_8 r_9},$
 $\overline{r_8 b_9}, \overline{b_8 r_9}, \overline{b_8 b_9}; \overline{r_9 r_{10}}, \overline{r_9 b_{10}}, \overline{b_9 r_{10}}, \overline{b_9 b_{10}}; \overline{r_{10} r_{11}}, \overline{r_{10} b_{11}},$
 $\overline{b_{10} r_{11}}, \overline{b_{10} b_{11}}; \overline{r_{11} r_{12}}, \overline{r_{11} b_{12}}, \overline{b_{11} r_{12}}, \overline{b_{11} b_{12}}; \overline{r_{12} r_{13}}, \overline{r_{12} b_{13}},$
 $\overline{b_{12} r_{13}}, \overline{b_{12} b_{13}}; \overline{r_{13} r_{14}}, \overline{r_{13} b_{14}}, \overline{b_{13} r_{14}}, \overline{b_{13} b_{14}}; \overline{r_{14} r_{15}}, \overline{r_{14} b_{15}},$
 $\overline{b_{14} r_{15}}, \overline{b_{14} b_{15}}; \overline{r_{15} r_{16}}, \overline{r_{15} b_{16}}, \overline{b_{15} r_{16}}, \overline{b_{15} b_{16}}; \overline{r_{16} r_{17}}, \overline{r_{16} b_{17}},$
 $\overline{b_{16} r_{17}}, \overline{b_{16} b_{17}}; \overline{r_{17} r_{18}}, \overline{r_{17} b_{18}}, \overline{b_{17} r_{18}}, \overline{b_{17} b_{18}}; \overline{r_{18} r_{19}}, \overline{r_{18} b_{19}},$
 $\overline{b_{18} r_{19}}, \overline{b_{18} b_{19}}; \overline{r_{19} r_{20}}, \overline{r_{19} b_{20}}, \overline{b_{19} r_{20}}, \overline{b_{19} b_{20}}; \overline{r_{20} r_{21}}, \overline{r_{20} b_{21}},$
 $\overline{b_{20} r_{21}}, \overline{b_{20} b_{21}}; \overline{r_{21} r_{22}}, \overline{r_{21} b_{22}}, \overline{b_{21} r_{22}}, \overline{b_{21} b_{22}}; \overline{r_{22} r_{23}}, \overline{r_{22} b_{23}},$
 $\overline{b_{22} r_{23}}, \overline{b_{22} b_{23}}; \overline{r_{23} r_{24}}, \overline{r_{23} b_{24}}, \overline{b_{23} r_{24}}, \overline{b_{23} b_{24}}; \overline{r_{24} r_{25}}, \overline{r_{24} b_{25}},$
 $\overline{b_{24} r_{25}}, \overline{b_{24} b_{25}}; \overline{r_{25} r_{26}}, \overline{r_{25} b_{26}}, \overline{b_{25} r_{26}}, \overline{b_{25} b_{26}}; \overline{r_{26} r_{27}}, \overline{r_{26} b_{27}},$
 $\overline{b_{26} r_{27}}, \overline{b_{26} b_{27}}; \overline{r_{27} r_{28}}, \overline{r_{27} b_{28}}, \overline{b_{27} r_{28}}, \overline{b_{27} b_{28}}; \overline{r_{28} r_{29}}, \overline{r_{28} b_{29}},$
 $\overline{b_{28} r_{29}}, \overline{b_{28} b_{29}}; \overline{r_{29} r_{30}}, \overline{r_{29} b_{30}}, \overline{b_{29} r_{30}}, \overline{b_{29} b_{30}}; \overline{r_{30} r_{31}}, \overline{r_{30} b_{31}},$
 $\overline{b_{30} r_{31}}, \overline{b_{30} b_{31}}; \overline{r_{31} r_{32}}, \overline{r_{31} b_{32}}, \overline{b_{31} r_{32}}, \overline{b_{31} b_{32}}; \overline{r_{32} r_{33}}, \overline{r_{32} b_{33}},$
 $\overline{b_{32} r_{33}}, \overline{b_{32} b_{33}}; \overline{r_{33} r_{34}}, \overline{r_{33} b_{34}}, \overline{b_{33} r_{34}}, \overline{b_{33} b_{34}}; \overline{r_{34} r_{35}}, \overline{r_{34} b_{35}},$
 $\overline{b_{34} r_{35}}, \overline{b_{34} b_{35}}; \overline{r_{35} r_{36}}, \overline{r_{35} b_{36}}, \overline{b_{35} r_{36}}, \overline{b_{35} b_{36}}; \overline{r_{36} r_{37}}, \overline{r_{36} b_{37}},$
 $\overline{b_{36} r_{37}}, \overline{b_{36} b_{37}}; \overline{r_{37} r_{38}}, \overline{r_{37} b_{38}}, \overline{b_{37} r_{38}}, \overline{b_{37} b_{38}}; \overline{r_{38} r_{39}}, \overline{r_{38} b_{39}},$
 $\overline{b_{38} r_{39}}, \overline{b_{38} b_{39}}; \overline{r_{39} r_{40}}, \overline{r_{39} b_{40}}, \overline{b_{39} r_{40}}, \overline{b_{39} b_{40}}; \overline{r_{40} r_{41}}, \overline{r_{40} b_{41}},$
 $\overline{b_{40} r_{41}}, \overline{b_{40} b_{41}}; \overline{r_{41} r_{42}}, \overline{r_{41} b_{42}}, \overline{b_{41} r_{42}}, \overline{b_{41} b_{42}}; \overline{r_{42} r_{43}}, \overline{r_{42} b_{43}},$
 $\overline{b_{42} r_{43}}, \overline{b_{42} b_{43}}; \overline{r_{43} r_{44}}, \overline{r_{43} b_{44}}, \overline{b_{43} r_{44}}, \overline{b_{43} b_{44}}; \overline{r_{44} r_{45}}, \overline{r_{44} b_{45}},$
 $\overline{b_{44} r_{45}}, \overline{b_{44} b_{45}}; \overline{r_{45} r_{46}}, \overline{r_{45} b_{46}}, \overline{b_{45} r_{46}}, \overline{b_{45} b_{46}}; \overline{r_{46} r_{47}}, \overline{r_{46} b_{47}},$
 $\overline{b_{46} r_{47}}, \overline{b_{46} b_{47}}; \overline{r_{47} r_{48}}, \overline{r_{47} b_{48}}, \overline{b_{47} r_{48}}, \overline{b_{47} b_{48}}; \overline{r_{48} r_{49}}, \overline{r_{48} b_{49}},$
 $\overline{b_{48} r_{49}}, \overline{b_{48} b_{49}}; \overline{r_{49} r_{50}}, \overline{r_{49} b_{50}}, \overline{b_{49} r_{50}}, \overline{b_{49} b_{50}}; \overline{r_{50} r_{51}}, \overline{r_{50} b_{51}},$
 $\overline{b_{50} r_{51}}, \overline{b_{50} b_{51}}; \overline{r_{51} r_{52}}, \overline{r_{51} b_{52}}, \overline{b_{51} r_{52}}, \overline{b_{51} b_{52}}; \overline{r_{52} r_{53}}, \overline{r_{52} b_{53}},$
 $\overline{b_{52} r_{53}}, \overline{b_{52} b_{53}}; \overline{r_{53} r_{54}}, \overline{r_{53} b_{54}}, \overline{b_{53} r_{54}}, \overline{b_{53} b_{54}}; \overline{r_{54} r_{55}}, \overline{r_{54} b_{55}},$
 $\overline{b_{54} r_{55}}, \overline{b_{54} b_{55}}; \overline{r_{55} r_{56}}, \overline{r_{55} b_{56}}, \overline{b_{55} r_{56}}, \overline{b_{55} b_{56}}; \overline{r_{56} r_{57}}, \overline{r_{56} b_{57}},$
 $\overline{b_{56} r_{57}}, \overline{b_{56} b_{57}}; \overline{r_{57} r_{58}}, \overline{r_{57} b_{58}}, \overline{b_{57} r_{58}}, \overline{b_{57} b_{58}}; \overline{r_{58} r_{59}}, \overline{r_{58} b_{59}},$
 $\overline{b_{58} r_{59}}, \overline{b_{58} b_{59}}; \overline{r_{59} r_{60}}, \overline{r_{59} b_{60}}, \overline{b_{59} r_{60}}, \overline{b_{59} b_{60}}; \overline{r_{60} r_{61}}, \overline{r_{60} b_{61}},$
 $\overline{b_{60} r_{61}}, \overline{b_{60} b_{61}}; \overline{r_{61} r_{62}}, \overline{r_{61} b_{62}}, \overline{b_{61} r_{62}}, \overline{b_{61} b_{62}}; \overline{r_{62} r_{63}}, \overline{r_{62} b_{63}},$
 $\overline{b_{62} r_{63}}, \overline{b_{62} b_{63}}; \overline{r_{63} r_{64}}, \overline{r_{63} b_{64}}, \overline{b_{63} r_{64}}, \overline{b_{63} b_{64}}; \overline{r_{64} r_{65}}, \overline{r_{64} b_{65}},$
 $\overline{b_{64} r_{65}}, \overline{b_{64} b_{65}}.$

3 运算符系统

许多学者认为,DNA 计算是利用生物学这个几乎是定性的学科来求解定量性的问题.似乎这个道理是正确的,因为目前的生物学虽然在量化的研究上取得了一系列的工作,但对整个生物学而言,定性的研究仍占绝大多数.但是,对于生物计算,确切地讲,对于 DNA 计算而言,在具体设计模型中,必须设计出不仅是量化的更是精确化的计算模型.我们在前面的几个研究中,发展了充分利用 PCR 技术进行量化研究的方法,在本模型中,仍将采用这一技术对“数据进行信息处理”.

从信息处理的角度,DNA 计算机应属于基因工程的范畴,而 PCR 技术是整个基因工程最为核心的技术,可以这样讲,没有 PCR 技术,就没有当前的基因工程.足见 PCR 技术在基因工程乃至整个生命科学、生物计算机科学中的重要性.

然而,PCR 在进行信息处理时,基本上属于定性的.定性的实验结果对于相同引物之间是否存在 DNA 序列可以准确地给出:有还是没有!但是,这两个引物之间经过 PCR 扩增之后存在多少条,一般很难给出准确的结果.也就是说,像 Adleman 在 1994 年给出的那样通过 PCR 技术将两个顶点之间的所有可能的路径全部扩增出来几乎是不可能的.这主要是因为两个引物之间的 DNA 序列的长度不等,且不同的 DNA 链数量很多.这也是许多学者对

Adleman 实验重复不出来的原因所在.

幸运的是,在两个引物之间,若扩增的都是等长的 DNA 序列,则理论上可精确地全部扩增出来,并且大量实验也证明了这一点. 我们已经采用了该方法对 61 个顶点的图的 3 着色问题进行了成功实验. 在本文的实验设计中,我们进一步利用此技术展开对 Ramsey 数 DNA 计算模型的研究.

下面给出具体的实验步骤.

建立初始解空间

在本节中,为实现该 DNA 计算模型,我们对初始解空间的建立进行研究.

建立的方法可用电子计算机来实现首步,也可直接用 DNA 分子的合成与删除方法进行,我们以拟完成的 $r(5,5)$ 图为例说明,假设我们要找具有 6 个顶点,既不含 K_5 又不含 N_5 的所有子图,有两种方法.

第 1 种方法是用电子计算机将既不含 K_5 又不含 N_5 所有子图对应的 0-1 序列找到,然后将每个 0-1 序列用 DNA 分子合成即可,这种方法用电子计算机所耗时间几乎是不考虑的. 电子计算机算出的标定的 6-阶不含 K_5 且不含 N_5 的图共有 25600 个,这个数字要合成是很麻烦的!事实上,我们在具体操作中只选择其中的若干个.

第 2 种方法是用生物技术来建立数据库. 即采用 DNA 自动合成仪合成初始 p -阶子图数据库. 目前,固相亚磷酸酰胺法是绝大部分 DNA 自动合成仪所使用的方法,其原理与步骤如下.

首先将欲合成的寡核苷酸链 3' 末端核苷(N_1)与固相载体偶联, N_1 的 5'-羟基用二甲氧基三苯甲基保护. 然后从 N_1 开始逐步地接长寡核苷酸链. 1 个合成循环包括 4 步,即保护、偶联、加帽和氧化反应. 每完成 1 轮循环,接长 1 个核苷酸. 接长的链始终被固定在不溶的固相载体上,过量的未反应物或分解物则通过过滤或洗涤除去. 当整个链达到预定的长度后,从固相载体上切下,脱去保护基,并经过分离纯化得到所需要的最后产物.

利用这种方法,我们可以合成行如式(5)所示的 DNA 链:

$$X = \{x_1 x_2 \cdots x_{65}; x_i \in \{r_i, b_i\}, i = 1, 2, \cdots, 65\} \tag{5}$$

以合成好的 DNA 链为模板,以 $\langle x_1, \bar{x}_{65} \rangle$ 为引物对,进行扩增,得到长度为 1300bp 的 DNA 序列集合即为初始解空间.

删除非解

在此仍以求解 $r(5,5)$ 为例给予说明运算子系

统是模型中的一个关键问题,我们主要通过以下步骤来实现对非解的删除.

第 1 步,从 X 中删去部分含有 K_5 的 DNA 链,即从上述合成的 DNA 链中删除包含有以下序列的 DNA 链:

$$\begin{aligned} \{1, 2, 3, 4, 5\} &= \{12, 13, 23, 14, 24, 34, 15, 25, 35, 45\} \\ &= \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}\} \\ \{1, 2, 3, 4, 6\} &= \{12, 13, 23, 14, 24, 34, 16, 26, 36, 46\} \\ &= \{l_1, l_2, l_3, l_4, l_5, l_6, l_{11}, l_{12}, l_{13}, l_{14}\} \\ \{1, 2, 3, 5, 6\} &= \{12, 13, 23, 15, 25, 35, 16, 26, 36, 56\} \\ &= \{l_1, l_2, l_3, l_7, l_8, l_9, l_{11}, l_{12}, l_{13}, l_{15}\} \\ \{1, 2, 4, 5, 6\} &= \{12, 14, 24, 15, 25, 45, 16, 26, 46, 56\} \\ &= \{l_1, l_4, l_5, l_7, l_8, l_{10}, l_{11}, l_{12}, l_{14}, l_{15}\} \\ \{1, 3, 4, 5, 6\} &= \{13, 14, 34, 15, 35, 45, 16, 36, 46, 56\} \\ &= \{l_2, l_4, l_6, l_7, l_9, l_{10}, l_{11}, l_{13}, l_{14}, l_{15}\} \\ \{2, 3, 4, 5, 6\} &= \{23, 24, 34, 25, 35, 45, 26, 36, 46, 56\} \\ &= \{l_3, l_5, l_6, l_8, l_9, l_{10}, l_{12}, l_{13}, l_{14}, l_{15}\} \end{aligned}$$

第 2 步,完成第 1 步后,我们对剩下的 DNA 链进行加位处理. 由于加位后会引入含有 K_5 或 N_5 的非解. 利用与上述相同的方法,对这些非解进行删除.

为实现该 DNA 计算模型中的运算子系统,我们研究了多种生物技术,考虑采用酶切技术或 PCR 等技术来实现删除非解这一操作.

(1) 酶切技术

酶切技术主要是利用限制性内切酶来切割 DNA 分子内部的磷酸二酯键,根据切割内容、切割位点以及切割方法而不同. 在 DNA 计算中,酶切技术已经被广泛应用. 本实验中拟采用酶切技术对非解进行删除. 以删除 K_5 为例,将选用 10 种限制性内切酶,通过这些酶的酶切反应来删除非解. 具体的操作方法,首先将得到代表解空间的 DNA 链平均分成 10 份,然后在每份溶液中加入对应的限制性内切酶,切断对应位上为 1 的 DNA 序列. 反应结束后,合并这些溶液,则这些溶液中不含有 10 条边同时为 1 的 DNA 序列. 也采用同样的方法可以删除对应位点为 0 或者 1 的 DNA 序列.

(2) PCR 技术

聚合酶链式反应(Polymerase Chain Reaction, PCR)是目前生物技术中应用最为广泛的 DNA 体外扩增法,也是在 DNA 计算中可以量化的技术. 关于 PCR 技术,可按照图顶点着色的方法进行处理.

(3) 综合技术

我们拟将多种生物技术手段综合考虑,或者采

用优化思想和 DNA 计算相结合的办法,将非解程序化删除.

当然,我们根据这里选用的不同的生化反应技术的选取,将能反馈至 DNA 计算的编码,根据每个特定的反应,建立综合性编码方案及适用于不同反应的编码约束条件.

4 检测子系统

经过运算子系统处理后,所剩下的 DNA 序列就可以代表可能的 Ramsey 图. 我们利用 DNA 测序

技术对所剩下的 DNA 链进行测序.

DNA 的序列测定是分子生物学领域的重要技术之一,是了解基因结构和功能的基础. 采用 DNA 计算有些时候也需要对 DNA 链进行序列分析,来判断变量的存在状态. 目前通用的两种 DNA 序列测定法有两种是 Maxam-Gilber 化学法和 Sanger 的双脱氧终止法,它们都是建立在分辨率极高的变性聚丙烯酰胺凝胶电泳基础上的. 这种电泳可将相差仅为一个核苷酸的单链 DNA 区分开来. 在实用中,往往以后者为主,其主要原理如图 4 所示.

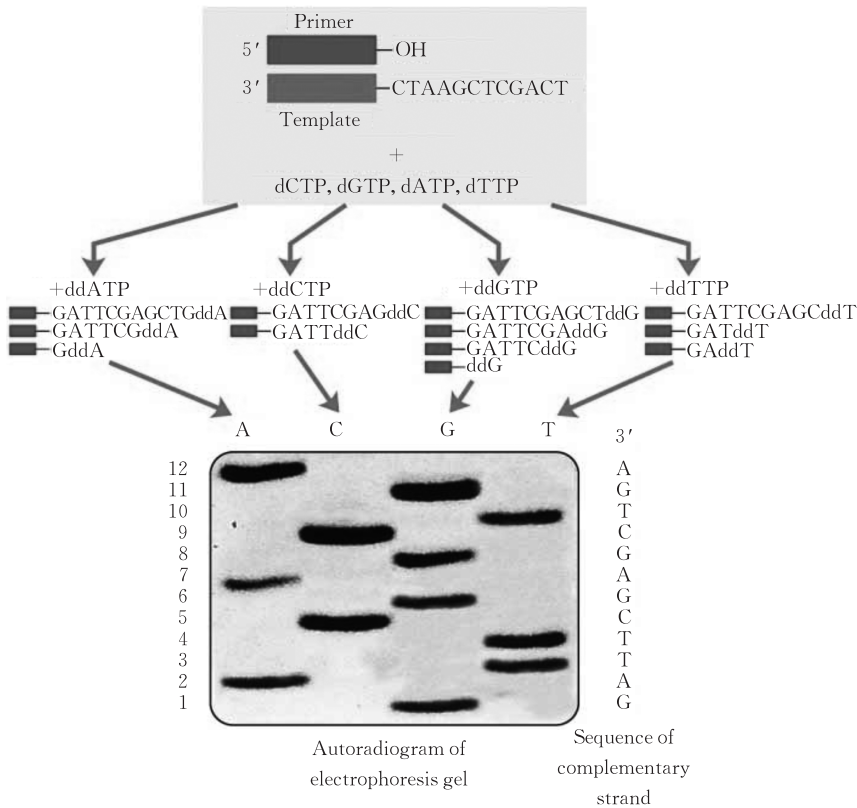


图 4 双脱氧链终止法测序示意图

最后,通过利用生物学实验仿真软件 Gene tool 分析测序结果,给出可能的 Ramsey 图.

5 总 结

求解 Ramsey 的一个常用方法就是构造 Ramsey 图,但是,由于图论理论的特殊性,如随着定点数 p 的增加,边数 $q = \frac{1}{2}(p-1)p$ 增长得较快,而其子图的数目以更快的速度增加,所以要穷举一个图的所有子图是非常困难的. 这里面还必须考虑图的同构问题,使问题变得更加复杂.

本文将 DNA 计算应用于 Ramsey 数的求解,提出了一种基于加位序列的 DNA 计算模型. 该模型在理论上对用 DNA 计算求解 Ramsey 数进行了理论研究,分析论证表明,该模型具有编码量少、求解过程简单、容易实现自动化等优点,可用于解决一定规模的 Ramsey 问题. 它丰富了 DNA 计算理论的研究,拓宽了 DNA 计算的应用领域.

该模型借鉴基于溶液技术的 DNA 计算模型,所用的分子生物学技术均为常规技术操作,包括 DNA 链的合成、限制性内切酶的酶切反应、PCR 扩增等,所有这些生物操作已经在 DNA 计算模型中得到广泛使用,并且其可行性均经过实验验证. 而

且,本文中提出的主要反应是酶切反应-PCR 技术以及测序技术,这些技术所需的实验条件都很成熟,且实验重复性很好.所以,尽管本文仅仅给出简单的图顶点着色问题的 DNA 计算模型,并没有进行生化实验验证,但在实验上是可行的.

当然,由于本文是首次将 DNA 计算应用于 Ramsey 数的研究,没有给出具体的实验操作,所有这些都有待于我们今后进一步研究和完善.

参 考 文 献

[1] Ramsey F P. On a problem of formal logic. *Proceedings of the London Mathematical Society*, 1930, 30: 264-286

[2] Graham R L, Rothschild B L, Spencer J H. *Ramsey Theory*. 2nd Edition. New York: A Wiley-Interscience Publication, John Wiley & Sons, 1990

[3] Radziszowski S P. Small ramsey numbers. *Electronic Journal of Combinatorics Dynamical Survey*, 2006, 11: 3-36

[4] Mckay B D, Radziszowski S P. $R(4, 5) = 25$. *Journal of Graph Theory*, 1995, 19: 309-322

[5] Mckay B D, Zhang K M. The value of the ramsey number $R(3, 8)$. *Journal of Graph Theory*, 1992, 16: 99-105

[6] Xu Jin, Fan Yue-Ke. Classical ramsey number DNA computing model (I): Add-bit-sequence model. *Chinese Journal of Computers*, 2008, 31(12): 2073-2080(in Chinese)

(许进,范月科. 经典 Ramsey 数 DNA 计算模型(I): 位序列计算模型. *计算机学报*, 2008, 31(12): 2073-2080)

[7] Adleman L M. Molecular computation of solution to combinatorial problems. *Science*, 1994, 266: 1021-1024

[8] Lipton R J. DNA solution of hard computational problems. *Science*, 1995, 268: 542-545

[9] Liu Q, Wang L, Frutos A G et al. DNA computing on surfaces. *Nature*, 2000, 403: 175-179

[10] Benenson Y, Paz-lizur T, Adar R et al. Programmable and autonomous computing machine made of biomolecules. *Nature*, 2001, 414: 430-434

[11] Braich R S, Chelyapov N, Johnson C, Rothmund P W K, Adleman L M. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 2002, 296: 499-502

[12] Benenson Y, Gil G, Ben-Dor U, Adar R, Shapiro E. An autonomous molecular computer for logical control of gene expression. *Nature*, 2004, 429: 423-429

[13] Ouyang Q et al. DNA solution of the maximal clique problem. *Science*, 1997, 278: 446-449

[14] Fan Yue-Ke. DNA computing models for classical Ramsey number problem[Ph. D. dissertation]. Huazhong University of Science and Technology, Wuhan, 2008(in Chinese)

(范月科. 求解 Ramsey 数的 DNA 计算模型[博士学位论文]. 华中科技大学,武汉,2008)

[15] Xu J, Qiang X L, Fang G et al. A DNA computer model for solving vertex coloring problem. *Chinese Science Bulletin*, 2006, 51(20): 2541-2549

[16] Zhang K, Pan L Q, Xu J. A global heuristically search algorithm for DNA encoding. *Progress in Natural Science*, 2007, 17(6): 745-749



XU Jin, born in 1959, professor, Ph. D. supervisor. His research interests include DNA computing and DNA computer, neural networks, genetic algorithms, graph theory etc.

FAN Yue-Ke, born in 1959, Ph. D. . His research interests include DNA computing and DNA computer, graph theory etc.