DNA计算：科研浅析

Lila Kari, and Kalpana Mahalingam

University of Western Ontario,
Department of Computer Science,
London, ON, N6A 5B7, Canada
lila, kalpana@csd.uwo.ca

1 引言

在最近的几十年中，我们从一个全新的角度见证了计算理论和实践中的令人兴奋的新发展。DNA计算，也被称之为分子计算、生物计算、或生物分子计算，是一个处于计算机科学和分子生物学交叉点上的领域。它是基于以下的想法：数据可以被编码到DNA片段上，并且分子生物学中的一些技术可以被用来进行数学和逻辑上的操作。

本章的主要目的是让读者简单了解在DNA计算中使用的一些技术和方法，并且概括介绍一些这个领域中的理论性的和实验性的研究。本章介绍以下内容：一个有代表性的DNA计算实验，一些自发的可编程的分子计算机的设计，和一些DNA存储器的模型。我们还介绍以下科研方向：在细胞中的计算，借助细胞的计算，DNA自我组装，一个可以用来计算或者产生DNA纳米机械的方法。这些DNA纳米机械可以是静态的或者动态的。本章的主要目的是为了引领读者进入一个令人震撼的DNA计算的世界，而不是为了做一个这个领域的全面回顾。

本章的组织如下。[1] 第2节介绍如何使用一些分子生物学中的符号来表示作为信息存储介质的DNA，并且介绍一些可以被用来操作DNA片段的生物操作。做为一个有代表性的DNA算法，Adleman和他的小组在试管

[1] 这篇文章的深度和广度是受到本章所针对的读者，我们的专业知识，以及本章的篇幅限制的影响。本章包含的研究课题并不是一个全面的列举，而是一个DNA计算研究的样本。此外，每个领域或者课题的长度是取决于一些因素，但是不代表它们的相应的重要性。

中完全使用生物操作解决了一个计算问题。这个计算问题是一个有20个变量的3-SAT问题的实例。我们在第3节中介绍这个实验。在第4节中，我们描述一个Benenson自动机，并以此来例证在自发的可编程的DNA分子计算机方面的研究，并且介绍它们在智能药物方面的潜在应用。在第5节中，我们讲解在DNA存储器方面的研究，比如嵌入式引物（primer）分子存储器（5.1节），有机的DNA存储器（5.2节），以及探讨为了DNA计算而进行的最优化的DNA序列的设计（5.3节）。第6节探讨一些在细胞中的（*in vivo*）计算提供给我们的潜在的机会，以及它可能给我们带来的对于自然界中运算机制的新的认识。在细胞中的（*in vivo*）计算指的是在细胞内或者借助细胞进行计算。在第7节中，我们介绍由自我组装（self-assembly）来进行DNA计算。它是DNA计算中最重要的研究方向之一，并且对快速发展的纳米技术有着重要的影响。此节介绍以下几个方面：自我组装的实验（7.1节），静态的DNA纳米结构（7.2节），以及给人深刻印象的DNA纳米机械（7.3节）。最后，在第8节，我们总结并且简单评述了最近在DNA计算领域的新发展。

2 分子生物基础

我们希望这章的读者是计算机科学家，并且了解一些生物学的知识。在这节里，我们提供一些对于理解本章内容所必要的生物学符号（DNA结构和DNA生物操作）。

作为遗传信息的载体，DNA（脱氧核糖核酸）存在于任何由细胞组成的有机体中[39]。DNA是一个由一些单体（monomer）组成的聚合体（polymer）。每一个单体是一个**核苷酸**（nucleotide）。每个聚合体也被称之为“多聚核苷酸”（polynucleotide）。更准确地说，DNA是一个由四种不同核苷酸组成的线性链状结构。每个核苷酸由一个碱基（腺嘌呤（Adenine），胞嘧啶（Cytosine），鸟嘌呤（Guanine），或者胸腺嘧啶（Thymine））和一个糖-磷酸盐单元(sugar-phosphate unit)组成。一些糖-磷酸盐单元被共价键(covalent bond)连接起来形成单链DNA的骨架。由于核苷酸之间的区别只是在于碱基的不同，一个单链DNA片段可以被看作是由四个不同字母组成的词（word）。这四个字母取自字母集{A, C, G, T}。根据他们的化学性质，每条单链DNA片段都具有方向性：一端被称为5’端，令一端被称为3’端。按照惯例，一个由DNA字母集中字母所组成的词表示一个在5’→ 3’ 方向上的单链DNA片段。也就是说，GGTTTTT表示的单链DNA片段是5’-GGTTTTT-3’。通常，一个短于20个核苷酸的单链多聚核苷酸被称之为**低聚核苷酸**

（oligonucleotide）。一个DNA片段有时被称作n-mer长，这里n代表它的长度，也就是它的核苷酸单体的数量。

单链DNA片段们的一个关键性的特点是它们的**Watson-Crick互补性**[91]：A与T互补，C与G互补。在不同方向上的互补的两个单链DNA片段将会结合在一起，形成一个稳定的双链结构。结合的原因是它们的碱基之间形成氢键。形成的双链结构是：骨架在外，结合的碱基对排列在内。在一个双链DNA片段中，两个位于相反片段上的互补的碱基被氢键结合起来形成了一个**碱基对**（bp）。

在下面这个例子中，两个DNA片段，5'-AAAAACC-3'和5'-GGTTTTT-3'，结合起来形成7个碱基对长的（7bp）双链片段

$$5' - \text{AAAAACC} - 3'$$
$$3' - \text{TTTTTGG} - 5'^{\circ}$$

正式地说，一个DNA片段w的Watson-Crick互补片段将会被表示为WK(w)或者$\overleftarrow{w}$。按照我们对于DNA片段方向性的惯例，我们应注意到WK(AAAAACC) = GGTTTTT。

另一种被计算使用的核酸是RNA[23]。**核糖核酸**（RNA）是一种和DNA相似的核酸，但是它们有以下三种主要区别：RNA通常是单链结构，而DNA通常是双链结构；RNA的核苷酸包含的是sugar ribose，而DNA的核苷酸包含的是sugar deoxyribose；尿嘧啶（Uracil）代替DNA中的胸腺嘧啶在RNA中出现。

染色体组由DNA序列组成。其中一些DNA序列是可以被转化为信使RNA（mRNA）的基因。这些基因根据遗传代码被转化为蛋白质。在转化的过程中，每个长度为3个字母的RNA片段（**密码子**（codon））被映射到一种氨基酸。一些特定的长度为3的片段被称之为开始（结束）密码子。它们表示一个转化的开始（结束）。一个蛋白质是由一些氨基酸组成的序列。这些氨基酸来自20种不同的氨基酸种类。

有很多**生物操作**可以被用来进行计算[2, 38, 64]，例如：基于Watson-Crick互补性的杂交结合（hybridization）；基于酶的剪切和粘贴操作；根据需要，合成最大为一个特定长度的DNA片段；以指数级的数量产生一个DNA片段的复制品；区分长度不同的片段；提取包含特定子片段（subsequence）的DNA片段；以及读出一个DNA片段。所有这些操作都已经被用来控制DNA计算和进行自动的DNA操作。

方向相反的单链DNA片段会结合生成一个双螺旋结构。这个过程被称之为**碱基配对**（base-pairing）、**降温结合**（annealing）、或者**杂交结合**（hybridization）。与之相反的过程——一个双螺旋结构分裂成两个单链结构

——被称作**熔解**（melting）。就像这些名字所表达的，熔解是通过提高温度达成的，降温结合是通过降低温度达成的。

一类酶被称之为**限制内切酶**（restriction endonucleases）。它们中的每一个都识别一个短的被称之为**识别点位**（restriction cite）的DNA片段。如果一个双链结构的DNA包含一个酶的识别点位, 那么它会被这个酶在这个位置切断。根据不同的酶，这个切断操作产生两种不同的结果：一种是产生两个“平滑端”（blunt-end），另一种更常见的是产生两个被称之为“黏端”（sticky-end）的悬垂的单链。另一种被称之为**连接酶**（ligase）的酶可以把两个不完全的双链DNA片段连接起来。这个连接是通过连接两个黏端骨架上的共价键完成的。这个过程叫做**连接反应**（ligation）。

凝胶电泳（gel electrophoresis）技术使我们可以按照长度区分DNA片段。带负电的DNA分子被放置在带有电场的湿凝胶的上端。电场会导致这些分子向湿凝胶的下端移动。较大的分子在凝胶中的移动速度较慢。经过一段时间，根据不同的长度，这些分子会被分在不同的亮带（band）中。

亲和提纯（affinity purification）可以在包含很多不同单链DNA片段的混合溶液中，提取带有特定子序列v的单链DNA片段。我们把一些合成的与v互补的片段附在带有磁场的小珠上。然后把混合溶液倒在这些小珠上。由于Watson-Crick互补性，那些带有v的片段会存留下来，而那些不包含v的片段不会存留下来。

DNA**聚合酶**（polymerase enzyme）的作用之一是通过**聚合酶链式反应**（PCR）来复制DNA。PCR复制反应需要一个单链DNA片段来做为**模板**，并且需要一个叫做**引物**（primer）的低聚核苷酸。这个引物会粘合到DNA模板上。DNA聚合酶会通过不断地在引物的一端增加核苷酸来催化DNA合成。引物会按照从5’端到3’端的方向在3’端被延长，直到获得所需要的片段。这个片段是由引物开始，并且与模板互补。如果我们使用两个引物，我们可以以指数级增长的速度来复制夹在两个引物之间的子序列。这个过程叫做**扩增反应**（amplification）。下面我们解释这个反应。出于解释的目的，我们使用以下符号。如果x是由字母集{A,C,G,T}中的字母组成的字符串，那么$\overline{x}$则表示与它互补的字符串，例如，

$$\overline{\text{AACCTTGG}} = \text{TTGGAACC}。$$

假设我们要扩增在双链DNA片段$\begin{matrix}5'-\alpha x\beta y\delta-3'\\ 3'-\bar{\alpha}\bar{x}\bar{\beta}\bar{y}\bar{\delta}-5'\end{matrix}$中，夹在$x$和$y$之间的部分。在这个片段中，$\alpha$、$x$、$\beta$、$y$、$\delta$表示DNA片段。接下来，我们使用$x$片段和与$y$互补的片段作为引物。在加热溶液并且这个双链DNA片段分解为两个单链之后，我们对溶液降温。与y互补的片段会粘合到上

面那个片段，x会粘合到下面那个片段。之后，聚合酶会在从5'端到3'端的方向上，延长这两个引物的3'端，并且产生两个不完整的双链片段：$\begin{matrix}5'-\alpha x\beta y\delta-3'\\ 3'-\bar{\alpha}\bar{x}\bar{\beta}\bar{y}\ \ -5'\end{matrix}$和$\begin{matrix}5'-\ \ x\beta y\delta-3'\\ 3'-\bar{\alpha}\bar{x}\bar{\beta}\bar{y}\bar{\delta}-5'\end{matrix}$。根据相似的机制，在下一个加热－冷却循环中，我们将得到另外两个片段$5'-x\beta y-3'$和$3'-\overline{x\beta y}-5'$。这两个片段是具有Watson-Crick互补性的。由于它们在以后的每一个循环中被复制，从这个循环起，它们的数量会超过其它片段的数量。最后，我们会得到数量级为2^n的，夹在x和y之间的子序列。这里，n是加热－冷却循环的数量。

其它的生物操作也都在生物计算中被采用。为了把信息编码到DNA上，我们可以设计一种编码机制来把原有的字母集映射到的只包含A、C、G、T的字符串上，然后再合成这些带有被编码的信息的单链DNA片段。一个生物计算由连续的生物操作完成[20]，例如，使用本节中介绍过的操作。做为生物计算输出的DNA片段可以被读出（使用测序器）并且解码。

3 Adleman的3-SAT实验

使用DNA作为一个计算机工作元素的想法可以追溯到1994年[1]。那时，Adleman解决了一个7节点的汉密尔顿路径问题的个例。他仅仅使用DNA来对信息编码，并且使用分子生物学中的技术作为算法的指令。本节介绍另一个更为复杂的DNA计算的实验。它解决了一个更大的3-SAT问题的实例[15]。

3-SAT问题是一种NP完全计算问题。目前已知的这类问题的顺序解法都需要指数级的时间。在Lipton[54]展示了这类问题可以充分利用分子计算提供的并行运算机制之后，它们成为了测试DNA计算机性能的平台。在2002年，Adleman和他的研究组解决了一个20个变量的3-SAT问题的实例[15]。和最初的概念验证性的DNA计算实验不同，这个实验第一次演示了DNA计算设备的计算能力超过了没有辅助设备的人。事实上，这个问题的答案是通过在一百多万个（2^{20}）可能的解中，彻底地搜索后找到的。

一个3-SAT问题的输入是一个由3个变量的与连接范式（conjunct normal form）（3CNF）组成的表达式，其中每一个与连接范式包含最多3个变量。如果存在一个真值赋值使一个表达式满足，那么，这个表达式被称作**可满足**（satisfiable）。也就是说，它的变量的真值赋值使这个表达式的结果为真。所以，如果存在这样的一个赋值，那么这个3-SAT的输出为“真”，否则为“假”。

Adleman使用的3CNF输入表达式包含了20个变量和24个分句：$\Phi=(\overline{x_3}\vee\overline{x_{16}}\vee x_{18})\wedge(x_5\vee x_{12}\vee\overline{x_9})\wedge(\overline{x_{13}}\vee\overline{x_2}\vee x_{20})\wedge(x_{12}\vee\overline{x_9}\vee\overline{x_5})\wedge(x_{19}\vee\overline{x_4}\vee x_6)\wedge(x_9\vee x_{12}\vee\overline{x_5})\wedge(\overline{x_1}\vee x_4\vee\overline{x_{11}})\wedge(x_{13}\vee\overline{x_2}\vee\overline{x_{19}})\wedge(x_5\vee x_{17}\vee x_9)\wedge(x_{15}\vee x_9\vee$

$\overline{x_{17}}) \wedge (\overline{x_5} \vee \overline{x_9} \vee \overline{x_{12}}) \wedge (x_6 \vee x_{11} \vee x_4) \wedge (\overline{x_{15}} \vee \overline{x_{17}} \vee x_7) \wedge (\overline{x_6} \vee x_{19} \vee x_{13}) \wedge (\overline{x_{12}} \vee \overline{x_9} \vee x_5) \wedge (x_{12} \vee x_1 \vee x_{14}) \wedge (x_{20} \vee x_3 \vee x_2) \wedge (x_{10} \vee \overline{x_7} \vee \overline{x_8}) \wedge (\overline{x_5} \vee x_9 \vee \overline{x_{12}}) \wedge (x_{18} \vee \overline{x_{20}} \vee x_3) \wedge (\overline{x_{10}} \vee \overline{x_{18}} \vee \overline{x_{16}}) \wedge (x_1 \vee \overline{x_{11}} \vee \overline{x_{14}}) \wedge (x_8 \vee \overline{x_7} \vee \overline{x_{15}}) \wedge (\overline{x_8} \vee x_{16} \vee \overline{x_{10}})$。其中，$\overline{x_i}$表示$x_i$的否定式，$1 \leq i \leq 20$。这个表达式$\Phi$是经过特殊设计的。它只有一个唯一的可满足的真值赋值。这个唯一的解是：$x_1 = F$，$x_2 = T$，$x_3 = F$，$x_4 = F$，$x_5 = F$，$x_6 = F$，$x_7 = T$，$x_8 = T$，$x_9 = F$，$x_{10} = T$，$x_{11} = T$，$x_{12} = T$，$x_{13} = F$，$x_{14} = F$，$x_{15} = T$，$x_{16} = T$，$x_{17} = T$，$x_{18} = F$，$x_{19} = F$，$x_{20} = F$。

Adleman的解是通过下面这个非确定（nondeterministic）算法得出的。

输入：由3CNF组成的布尔表达式Φ。
第一步：产生一个包含所有可能的真值赋值的集合。
第二步：删除所有使第一个分句为假的真值赋值。
第三步：对于输入表达式中所有的分句，执行第二步。
输出：最终留下的真值赋值（如果存在）。

为了实现这个算法，我们首先需要把输入信息编码到DNA片段中。这是由下面的过程实现的。每一个变量x_k，$k = 1, \ldots, 20$，对应两个不同的长度为15-mer的被称之为“值序列”（value sequence）的单链DNA片段。其中，被写作X_k^T的一个表示真（T），而被写作X_k^F的另一个表示假（F）。

下面是一些在试验中合成的，并且被使用的长度为15-mer的序列（它们都不包含核苷酸G）：

$$X_1^T = TTACACCAATCTCTT, \quad X_1^F = CTCCTACAATTCCTA,$$
$$X_{20}^T = ACACAAATACACATC, \quad X_{20}^F = CAACCAAACATAAAC。$$

在2^{20}种可能的解中，每一个解都被一个长度为300-mer的“库片段”（library strand）所表示。这些库片段是由对应变量的长度为15-mer的序列，按照顺序连接组成的。例如，下面这个片段

$$\alpha_1\alpha_2 \ldots \alpha_{20}, \quad 其中，\alpha_i \in \{X_i^T, X_i^F\}，\ 1 \leq i \leq 20。$$

为了得到这些“库片段”，40个长度为15-mer的片段（在溶液中，每一个片段有很多的复制品）通过使用混合并匹配的组合合成技术（mix-and-match combinatorial synthesis technique）[23]被连接起来了。

这个生物计算使用以下设备：一个装有“库模块”的玻璃杯，其中装有包含库的凝胶；以及24个装有“分句模块”（clause module）的玻璃杯，每一个玻璃杯对应一个表达式中的分句。每一个分句模块的玻璃杯装有包含取样器

（被固定的单链DNA片段）的凝胶。这些设计好的取样器只会和特定的库片段结合。这些特定的库片段代表那些满足分句的真值赋值。

这些片段在凝胶电泳的作用下在模块中间移动。也就是说，在电流的作用下，带有负电荷的DNA片段会在凝胶中移动。

实验开始时，整个库通过第一个“分句模块”。这样，库中那些带有使第一个分句满足的真值赋值的片段会被那些固定的取样器捕获，而那些不能满足第一个分句的片段会移动到一个收集器中。那些被捕获的片段可以通过提高温度被释放出来，然后作为第二个分句模块的输入。按照这样的方法，最终，只有那些满足24个分句的片段才能被保留下来。

作为输出的片段的数量被PCR扩增。所使用的引物对有4种情况。他们对应对第一个变量x_1和最后一个变量x_{20}赋值的4种方式。只有其中一对$(X_1^F, \mathrm{WK}(X_{20}^F))$产生了亮带。这就说明$x_1 = F$和$x_{20} = F$是满足条件的赋值。这个过程对每一对变量$(x_1, x_k)$重复，$2 \leq k \leq 19$，并且，根据观测到的亮带的长度，我们对这些变量赋值。这些由实验得到的值与这个表达式的解一致。由此，这个实验就完成了。

我们可以使用简单的生物操作来进行复杂的计算。例如，解决一个3-SAT问题的较大的实例。这是DNA计算一个出色的特点。

4 Benenson自动机

在Adleman[1]对DNA计算进行了概念验证性的演示之后，一些研究组希望找到一种通用图灵机（Universal Turing Machine）的DNA实现。在这个过程中，他们展示了多种自发的可编程的生物计算机[10, 11, 12, 56, 74]。这些计算机使用分子形式的输入和输出。在本节里，我们展示一种这样的计算机[10]。它被称之为“Benenson自动机”。

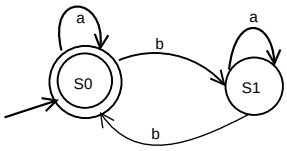

Fig. 1. 一个基于字母集$\{a, b\}$的，带有状态S_0和S_1的自动机。它只接收带有偶数个字母b的字符串。

在[10]中，作者们使用双链DNA分子和限制酶（restriction enzyme）建立了一个简单的含有两个状态的自动机，如图1。这个自动机的字母集由两个字母组成。

对于任何含有两个状态的自动机，它的转移规则（transition rule）只有8种可能，并且，它的接受状态的集合是它的双状态集合的一个子集。Benenson等实现的自动机读取的输入是由集合$\{a, b\}$中字母组成的字符串，并且，它只接受包含偶数个字母b的输入（图1）。

Benenson自动机中主要使用的是*FokI*酶。它属于一类特殊的酶。这类酶识别一个特定的DNA序列，并且在离这个序列不远的地方进行切割。*FokI*酶与下面这个序列结合

$$\begin{matrix} 5'-\text{GGATG}-3' \\ 3'-\text{CCTAC}-5' \end{matrix},$$

并且，（不取决于这个序列的构成）在离这个序列9bp的地方切开上面的DNA片段，在离这个序列13bp的地方切开下面的DNA片段。留下一个4个字母长的黏端。

输入字符a和b，以及结束符t，被编码为长度为6bp的序列：

$$\mathbf{a}: \begin{matrix} 5'-\text{CTGGCT}-3' \\ 3'-\text{GACCGA}-5' \end{matrix} \quad \mathbf{b}: \begin{matrix} 5'-\text{CGCAGC}-3' \\ 3'-\text{GCGTCG}-5' \end{matrix} \quad \mathbf{t}: \begin{matrix} 5'-\text{TGTCGC}-3' \\ 3'-\text{ACAGCG}-5' \end{matrix}.$$

在图3的第一行，我们可以看到一个对输入字符串编码的例子。输入字符串ab被编码为一个双链DNA片段。它首先包含一个*FokI*的识别点位，然后是字符串abt的DNA编码。

每一个状态/字母组合（state/symbol pair）被编码为一个长度为4-mer的DNA片段。状态/字母组合S_0a被编码为$5'-\text{GGCT}-3'$（字母a的、编码长度为4-mer的后缀），状态/字母组合S_1a被编码为$5'-\text{CTGG}-3'$（字母a的、编码长度为4-mer的前缀）。对于b和t的状态/字母组合的编码是相似的。这种方法可以使一个字母的编码以两种方式解释：如果一个字母的长度为4-mer的后缀被检测到，那么，我们可以看作，在状态S_0中读取了这个字母；如果一个字母的长度为4-mer的前缀被检测到，那么，我们可以看作，在状态S_1中读取了这个字母。

我们有两个用来检测输出的分子。S_0-D是一个长度为161-mer的双链DNA片段。它带有一个悬垂端$3'-\text{AGCG}-5'$，并且“检测”运算的最终状态是否为S_0。S_1-D是一个长度为251-mer的双链DNA片段。它带有一个悬垂端$3'-\text{ACAG}-5'$，并且检测运算的最终状态是否为S_1。由于这两个用来检测输出的分子长度不同，它们可以在凝胶电泳中被区分出来。

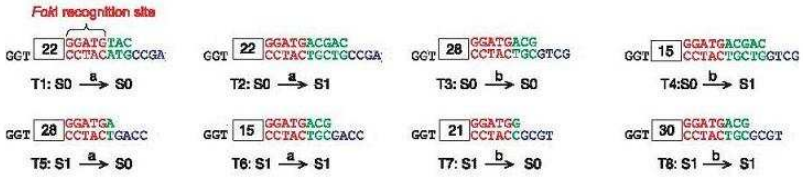

Fig. 2. 以分子形式编码的双状态自动机的所有可能的状态转化。正方形表示双链DNA片段。这些双链DNA片段的长度由正方形里面的数字表示。（这个表示方法取自Benenson, Y. et al., Nature, 414, 430, 2001.）图中的文字：*FokI* recognition site（*FokI*的识别点位）。

图2中显示了两个状态的Benenson自动机的八种可能的状态转换，这些状态转换由分子表示。每个分子带有一个4个字母长的悬垂端。例如，就像在下面的例子中具体解释的一样，在T_1中的$3'-\mathrm{CCGA}-5'$可以选择性地与代表当前状态/字母组合的DNA片段结合。

图3描述了这个自动机处理输入字符串ab的编码的过程。首先，*FokI*酶找到它的识别点位，并且切割输入字符串abt的编码，导致4个核苷酸长的黏端$5'-\mathrm{GGCT}-3'$被暴露出来。这个黏端表示S_0a这个状态/字母组。

代表$S_0a \to S_0$这个转移规则的分子T_1“检测”到这个状态/字母组。这个检测的完成，是通过T_1与被切割的输入分子结合，并且，在连接酶的帮助下，形成一个完整的双链DNA分子。

与新的分子结合的T_1包含着*FokI*酶的识别点位。此外，在*FokI*酶的识别点位之后的3bp长的占位序列保证下一次*FokI*酶的切割使代表下一个输入字母b的片段的后缀暴露出来。这个后缀代表S_0b。[2]

目前的悬垂端是$5'-\mathrm{CAGC}-3'$。它代表S_0b。这个黏端对应代表转移规则$S_0b \to S_1$的分子。由此，当前的分子和T_4在连接酶的帮助下形成另一个完整的双链DNA片段。

在当前分子上再一次使用*FokI*酶。这会使悬垂端$5'-\mathrm{TGTC}-3'$暴露出来。这个悬垂端是终结符的前缀，并且表示S_1t。这个悬垂端互补于检测分子S_1-D的黏端$3'-\mathrm{ACAG}-5'$。这个检测分子意味着运算的最后一个状态是S_1。状态S_1不是一个接受状态，所以，这个运算的结果是：输入ab不被这个自动机接受。

[2] 需要注意的是，如果这个自动机使用的是代表$S_0a \to S_1$这个转移规则的分子T_2，那么，T_2中的占位序列的长度应该为5bp，这使得切割的位置向左移动2bp。这将使代表字母b的片段的前缀被暴露出来。这个前缀表示S_1b。

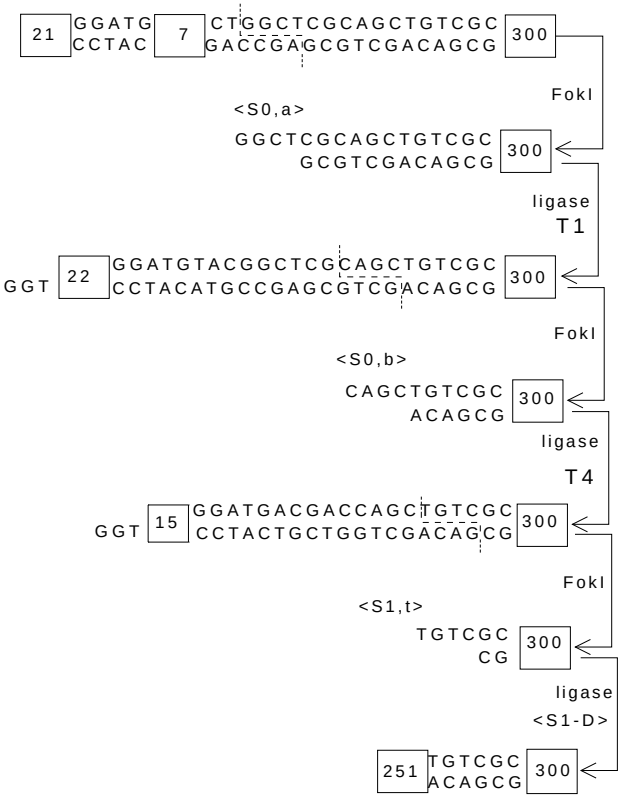

Fig. 3. 对应图1中自动机的Benenson自动机在输入字符串ab上的计算。正方形表示双链DNA片段。这些双链DNA片段的长度由正方形里面的数字表示。（这个表示方法取自Benenson, Y. et al., Nature, 414, 430, 2001.）图中的文字：ligase（连接酶）。

需要注意的是，上面这个方法可以创建任何含使用两个字母的双状态自动机。通过使用代表不同转移规则的分子，这些自动机可以执行不同的工作。

作为一个令人兴奋的应用，Benenson等[12]演示了如何使用上面这个方法来进行医学诊断和治疗。作为一个原理上的证明，Benenson等[12]设计了一个自动机来识别和分析与小细胞肺癌和前列腺癌相关的基因的mRNA，并且制造了基于抗癌药物建模的单链DNA分子。

这个自动机的计算过程被正式地写在[86]中。[86]的作者研究了Benenson自动机的计算能力，并且证明了这些自动机可以计算任何布尔函数。

5 DNA存储器

本节讨论一些用DNA进行信息存储的方法。在对一些DNA存储器做简要介绍之后，我们具体描述嵌入式引物分子存储器（NPMM）（5.1节），和在细胞中的（*in vivo*）有机的DNA存储器（5.2节）。此外，5.3节将讨论几种对于DNA计算中使用的、作为信息存储的、DNA序列的优化方法，其中包括软件虚拟、算法的方法、和理论研究。

有一些理由使我们来考虑使用DNA存储器作为现在使用的存储器的替代品。首要的原因是DNA可以提供非常高的信息存储密度。确实，DNA计算最吸引人的特定就是它提供了在很小的空间里存储大量信息的可能。在[71]中，Reif报告了基于下面因素的计算：一摩尔的DNA包含6.02×10^{23}个碱基单体，一个单体的平均分子重量是350克/摩尔。所以，1克的DNA包含2.1×10^{21}个碱基。由于4种DNA碱基可以对2位信息进行编码，1克DNA可以存储大约4.2×10^{21}位的信息。因此，DNA存储信息的潜力是传统存储技术的10^{10}倍。此外，DNA信息的稳定性可以确保信息被保存很长时间[4, 18, 85]。

Baum在[8]中首次提出了可对内容寻址的（content addressed）DNA存储器的想法。这个存储器可以存储固定长度的二进制词。在他的模型中，每一个在存储器中词是一个按照下面方法构造的单链DNA片段。两个不同的DNA序列被付给每一个组成部分（component）。其中，一个代表“1”，另一个代表“0”。对于一个特定的二进制词的DNA分子编码是由一些DNA序列按照任何顺序连接组成的。每一个序列对应这个词中的一个位。这样的DNA存储器是可结合的（associative），并且是可对内容寻址的。给出一个组成部分值的子集，我们可以像下面这样，在DNA存储器中检索任何带有这些值的词。对于每一个组成部分，我们可以合成与它互补的DNA子序列，并且把它粘贴在一个固定的支撑物上。例如，磁珠。这个片段会与DNA存储器中包含与之互补的片段的分子结合。这些分子可以被磁性提取出来。在重复提取每一个组成部分之后，我们得到了完全符合条件的DNA分子。就像在[8]中计算的一样，按照这种原理设计的存储器是非常惊人的。我们可以设想建造比大脑的存储容量大很多的存储器。

其他一些作者[16, 46, 60, 71, 88]也提出了一些DNA存储器的模型。在这些研究中，多数研究都包含了一些小规模的前期实验。最近，Yamamoto等提出了一个带有一千万个地址的DNA存储器[99]。他们提出的DNA存储器是通过嵌入式PCR寻址。我们在5.1节中具体讨论这个被称为NPMM的存储器。

5.1 嵌入式引物分子存储器

NPMM[99]是一个包含DNA片段的库。其中，每一个片段都包含数据信息和它所对应的地址信息。数据信息是通过一种合适的方式，编码为基于DNA字母集{A,C,G,T}的DNA序列。地址信息由几部分组成，并且也被编码为DNA序列。这些表示地址的序列被放置在数据序列的两端。例如，数据可以被储存为下面的形式

$$[CL_i, BL_j, AL_k, \text{数据}, AR_q, BR_r, CR_s]$$

其中，$i, j, k, q, r, s \in \{0, \ldots, 15\}$，并且，每一个地址的组成部分，例如，$CL_0$，表示一段长为20-mer的DNA序列。为了提取这个数据，我们使用由三个步骤组成的嵌入式PCR。在第一步中，我们使用PCR和引物对(CL_i, WK(CR_s))。结果是：我们可以得到在存储溶液中，所有的以CL_i开始并且以WK(CR_s)结束的DNA分子。这是由于PCR可以使被扩增的分子（以CL_i开始，并且以WK(CR_s)结束）的浓度远远大于那些没有被扩增的分子的浓度。后者可以在所有的实际情况中被忽略。在第二步中，我们使用PCR和引物对(BL_j, WK(BR_r))。目前，我们得到夹在CL_iBL_j和BR_rCR_s中被编码为DNA的数据。在第三步中，我们使用PCR和引物对(AL_k, WK(AR_q))。这使我们得到左面地址为$CL_iBL_jAL_k$，并且右面地址为$CR_qBR_rCR_s$的数据。这个分子的序列可以被读出，并且被解码。这样我们就可以读取被编码在DNA中的数据。

前面介绍的NPMM模型可以实现很大的地址空间和很高的特异性。这种分等级的地址结构允许我们使用少量的DNA序列来表示非常大的地址空间。在PCR的过程中，突变是可能发生的。这是NPMM主要的不利因素之一。这样的问题可以通过合适地选择DNA序列来避免。我们会在5.3节中具体讨论这个方法。在[99]中，作者们通过基于组合优化的理论模型和一些实验上的制约因素，讨论了扩大这种DNA存储器的极限。结果显示，这种DNA存储器模型的地址空间是接近于理论极限的。

5.2 有机的DNA存储器

为了长时间地保存数据，利用有机体作为DNA存储器比其它类似的存储器有更大的潜力[97, 98]。一个裸露的DNA很容易在开放的环境里被破坏，而一个活的有机体可以作为一个宿主来保护带有信息编码的DNA。在[97]中，作者们提出了一个活的有机体作为DNA存储器序列宿主的候选者。这个候选者可以携带人造的基因序列，并且在恶劣的条件下生存。

这个实验有几个关键性的阶段[97]。在第一个阶段中，在寻找携带嵌入DNA分子的候选者的过程中，作者们考虑了几种微生物，并且选择了两种被全面理解的细菌：**大肠杆菌**（Escherichia coli）和**耐辐射球菌**（Deinococcus radiodurans）。耐辐射球菌可以在极端的条件下存活。这包括寒冷、脱水、真空、酸性条件、和辐射。所以，它被称之为极端微生物（polyextremophile）。

在信息编码的阶段，一种特定的方案被选取出来。它把长度为3-mer的序列赋值给一些字母。例如，AAA代表数字“0”，AAC代表数字“1”，AGG代表英语字母“A”。每一个编码序列只包括4个DNA字母中的三个。通过使用这样的编码方式，任何英语文档都可以被编码为DNA序列。在这个实验中使用的文档是“And the oceans are wide”。

此外，为了以后检索和提取信息，一些DNA序列作为“标记”来标注被编码信息的开端和结尾。这些序列是通过搜索大肠杆菌和耐辐射球菌的基因组得到的。它们是一组固定长度的DNA序列。它们不存在于这些细菌的基因组，但是它们满足这些细菌的基因组的限制条件。25个这样的DNA序列被选出。它们的长度是20bp。它们不会引起不必要的基因突变，或者对细菌产生破坏。所有的序列都包含多个停止密码子TAA、TGA、和TAG作为子序列。没有这些停止密码子的存在，细菌会错误地解释这些存储体片段，把它们转化为mRNA，并且，把这些mRNA转换为可以破坏嵌入信息的、甚至杀死细菌的人造蛋白质。

一个长度为46bp的DNA序列被构建。它由两个长度为20bp的不同的“标记”序列、以及一个连接“标记”序列的长度为6bp的序列组成。这个长度为6bp的序列是一个特定的酶的识别点位。这个双链DNA分子被克隆到一个重组质粒（recombinant plasmid）里面（一个环形的双链的DNA序列）——如图4。

这个被嵌入的DNA接下来被嵌入到克隆载体（cloning vectors）（可以在细菌宿主中自我复制的环形DNA分子）中。得到的克隆载体又通过高伏特电击被转移到大肠杆菌中。这时，这些载体就可以进行自我复制。

此后，这些载体和带有编码的DNA又被合并到耐辐射球菌的基因组中，用来进行永久的信息存储和提取。使用已知的、在两端上的序列，信息可以被提取、读出、并且解码为原来的英语文档。

这个有机的DNA存储器有着巨大的信息存储的潜力，尤其是在考虑到1mL的液体中可以包含10^9个细菌。潜在的不利因素是随机的基因突变，但

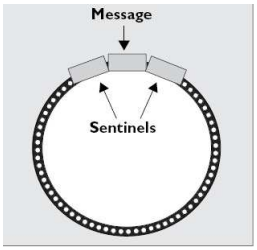

Fig. 4. 一个带有“标记”片段的重组质粒。这些“标记”片段是用来保护其间被编码为DNA的信息。（取自Wong, P. et al., Commun. ACM, 46, 95, 2003.）图中文字：Message（信息），Sentinels（标记）。

是，在细胞中自然存在的、对于错误检测和改正的机制下，这些基因突变不太可能发生。

5.3 DNA序列设计

在大多数基于DNA的计算中，有三个基本的阶段：（1）把输入数据编码到单链或者双链DNA分子上，（2）使用生物操作进行生物计算，（3）对结果解码。在DNA中，一个最主要的问题存在于第一阶段和对于用于信息编码的低聚核苷酸的优化设计。优化的设计可以减少Watson-Crick互补配对时的错误，并且避免DNA片段间不需要的结合。

在试验中，碱基的互补性确实可能引起潜在的问题。这是由于一些DNA片段形成非确定性的杂交结合，并且不完整地结合到与它们不完全互补的片段上，或者，由于它们以一种没有计划到的方式与自身或者其它片段结合。有一些方法被用来处理这些**序列设计问题**。在本节中，我们讨论几种被用来优化地设计在DNA计算中所使用的DNA片段的方法。其中包括：软件模拟的方法、算法的方法、和理论的方法。

软件模拟

在一个实验进行前，软件模拟是被用来对这个试验计划的正确性进行验证。一些软件包是为了DNA计算而编写的[24, 25, 32, 33]。例如，软件包*Edna*模拟在试验中出现的生化过程和反应。

Edna[28]是一个模拟软件。它使用一组计算机，并且模拟在试管中可能发生的过程。它可以被用来决定一个特定的编码方案是否合适，测试一个实验计算，并且估计这个实验计划的性能和可靠性，甚至可以评估这个试验计

划的复杂性。试管中的操作都被赋予一个基于很多反应条件的值。*Edna*对于复杂性的测量方式是累计一个试验计划中所有操作的值。这个软件提供的其它功能是：在多种反应条件下，预测DNA的熔解温度（这个温度是一个双链DNA片段“熔解”为两个单链DNA片段时的温度）。这个软件模拟的分子反应都是局部的，并且，它显示了分子生物反应过程中固有的随机性。

算法的方法

在大多数的DNA计算中，有这样一个假设：一个片段只和与它完全互补的片段结合。例如，DNA计算的结果被与之互补的片段从试管中提取。

然而，在实际中，一个DNA分子会和另一个片段结合。这个片段不同于与之完全互补的片段。其间的区别可以是一个、甚至是更多个核苷酸的区别。一种避免这个问题的方法是：保证溶液中任何两个分子在至少一个位置上有不同的碱基。这个性质可以被海明距离（Hamming distance）正式地描述。

对于两个长度相同的词w_1和w_2，它们的海明距离$H(w_1, w_2)$是这两个词中不相同的位置的数量。对于一个词的集合，基于海明距离的限制条件是：这个集合中的任何两个词w_1和w_2的海明距离大于d，$H(w_1, w_2) \geq d$。如果我们使用DNA词，另一个避免错误结合的必要的限制条件是$H(w_1, \mathrm{WK}(w_2)) \geq d$。这里，$\mathrm{WK}(w)$表示$w$的Watson-Crick互补片段。另一个需要考虑的方面是，当从溶液中提取运算结果时，杂交结合需要对于溶液中所有的分子同时发生。这意味着，对于所有正在发生的杂交结合反应，溶液温度应该是可以比较的。这就是在产生这个词的集合时所需要考虑的第三个主要的限制条件。

为了解决按照这三个限制条件设计DNA代码的问题，在[89]中，作者们提出了一个算法。这个算法是一个随机的局部搜索。熔解温度的限制条件被简化为下面这个条件。对于每一个片段，核苷酸C和G的数量是这个片段中核苷酸数量的一半。这个算法产生一个集合。这个集合包含长度相同的DNA序列，并且满足基于海明距离的、以及温度的限制条件。下面是这个算法：

输入：需要产生的词的数量k，以及这些词的长度n。
第一步：产生一个随机的集合，其中包含k个长度为n的词。
第二步：修改这个集合使它满足第一个限制条件。
第三步：对于所有的限制条件，重复第二步。
输出：词的集合（如果存在）。

更具体地说，在第二步中，我们从违反至少一个限制条件的集合中选出两个词w_1和w_2。在$1-\theta$的概率下，两个词中的一个被随机地改变一个碱基。

这里，θ是一个噪音参数。在这个过程中，我们需要最大化地减少被违反的条件的数量。这个算法有两个终止条件：一个是，这个集合不违反任何限制条件；另一个是，循环的执行次数超过了一个最大的阈值。经验证明，这个算法是有效的，并且，根据经验，对于任何问题，这个噪音参数的最优化的值是0.2。

理论研究

在这节里，我们讨论在[35]中介绍的使用形式语言（formal language）的理论方法来解决代码设计的问题。我们首先回顾一些概念和符号。一个字母集是一个有限的非空的字母的集合。Σ表示一个任意的字母集。那么，Σ^*则表示所有由这个字母集中字母组成的词，其中包括λ这个空词（empty word）。Σ^+包括所有Σ^*中非空的词。Σ^i包含所有由这个字母集中字母组成的长度为i的词。我们用$\mathrm{Sub}_k(L)$来表示由集合L中所有词的长度为k的子词组成的集合。对于形式语言理论（formal language theory）、码论（theory of codes）、和词的组合理论（combinatorics on words）中的具体内容，参考[14, 34, 55, 75, 83, 102]。

每一个和DNA有关的生物分子的实验都会生成核苷酸序列。这些序列构成一个基于字母集$\Delta = \{\mathrm{A,G,C,T}\}$的语言（language）。核苷酸的Watson-Crick互补性定义了一个自然的回旋（involution）θ：A$\rightarrow$T，G$\rightarrow$C。这个回旋是一个对于Δ的反态射（antimorphism）。一个回旋θ是这样的一个映射：θ^2是恒等式（identity）。一个反态射是这样定义的：对在Δ^*中所有的词u和v，$\theta(uv) = \theta(v)\theta(u)$。例如，如果$u$ =AGACT，那么它的Watson-Crick互补词$\theta(u)$是AGTCT。

对于一个DNA语言，也就是一个基于DNA字母集的语言，如果这个语言满足一些特定的性质，那么，它的两个词之间的不想要的Watson-Crick结合就可以被避免。有两种不想要的杂交结合：分子内的（intramolecular），和分子间的（intermolecular）。当一个DNA片段内的两个序列相反互补（如图5），分子内的杂交结合就会发生。在这种情况下，这个DNA片段会形成一个发夹结构（hairpin）。

假设我们要避免如图5所示的杂交结合。如果对于所有的词$w \in \Sigma^k$，L满足条件$\Sigma^* u \Sigma^m \theta(u) \Sigma^* \cap L = \emptyset$，那么，这个语言$L$被叫作$\theta$-$k$-$m$-子词码（$\theta$-$k$-$m$-subword code）（Jonoska等[36]）。一个相似的对于无发夹语言（hairpin-free language）的定义在[41]中被提出。在这个定义中，Σ是一个任意的字母集，并且，θ是一个任意的反态射的回旋。如果使用DNA字母集Δ作为Σ，那

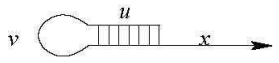

Fig. 5. 分子内的杂交结合。一个$uv\tilde{u}x$ 类型的DNA序列会形成一个叫作发夹结构的次级结构（secondary structure）。其中，$\tilde{u}$表示u的Watson-Crick互补片段。这样的发夹结构可以被无发夹语言和θ-k-m-子词码避免。

么这个性质实际上是意味着集合L中的DNA序列不会形成图5中这样的发夹结构。

在[36]中，作者们介绍了一些DNA片段的集合。这些被叫做θ-k-码（θ-k-code）的集合可以避免所有不想要的分子间的结合（如图6）。对于一个语言L，如果对于所有在$\mathrm{Sub}_k(L)$中的词$x, y \in \mathrm{Sub}_k(L)$，$\theta(x) \neq y$成立，那么，这个语言$L$是一个$\theta$-$k$-码。$\theta(x) = y$这个关系意味着$x$和$y$对应的分子可以在它们之间产生如图6所示的化学键。如果使用一个合适的k，那么，一个θ-k-码可以避免所有种类的不想要的分子间的杂交结合。

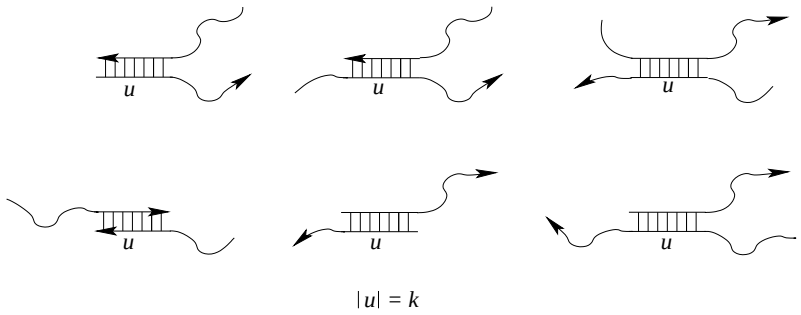

Fig. 6. 分子间的杂交结合。如果一个DNA分子包含一个长度为k的子词u，并且另一个DNA分子包含子词$\tilde{u}$，那么这个两个分子会结合起来产生图中的结构。这样的杂交结合可以被θ-k-码避免。

θ-k-码的性质是为了确保，在DNA计算中，DNA片段不形成不想要的杂交结合。根据这个性质设计的集合已经在实际的实验室中被成功地测试了[36]。在[44]中，θ-k-码的概念被扩展到海明无结合性质（Hamming bond-free property）：对于所有的在$\mathrm{Sub}_k(L)$中的词$x, y \in \mathrm{Sub}_k(L)$，$H(\theta(x), y) > d$成立。其中，$d$是一个根据经验的正整数参数。

假设我们使用的码带有以上介绍的属性，这个属性可能随着时间逐渐地在计算中消失。这个问题引领我们进入另一个研究领域。这个领域研究生物操作如何影响不同的无结合属性（bond-free properties）。这些操作包括剪切、粘贴、剪切、根据上下文的插入、以及删除。在这些操作下的恒定性在[36, 37, 40]中得到研究。Marathe等在[57]中研究了一些码（code）的大小的边界值。这些码满足一些可以被构建的性质。

通过建立一个用来描述和分析DNA信息编码的数学框架，本节描述的DNA信息编码的方法和它的一些性质。这些对于DNA计算的实验和理论研究都是有意义的。此外，仅从计算机科学的角度，在本节中定义的、以及研究的概念已经被证实为硕果累累。这些概念中的一些归纳了一些在编码理论和词的组合理论中的经典概念。比如：基本性（primitive）、交换性（commutative）、共轭性（conjugate）、倒序词（palindromic words）、前缀码（prefix code）、后缀码（suffix code）、中缀码（infix code）、以及无逗号码（comma-free code）（例如，见[19, 35, 40, 42, 43]）。

6 在细胞中的计算

在细胞中、以及利用细胞进行运算的研究是把自然界理解为一种计算模型的尝试。这种计算通常被叫做细胞计算，或者在细胞中的（*in vivo*）计算。这个领域中的一个研究方向所关心的是：在一种叫做纤毛虫（ciliates）的单细胞有机体中，基因组装（gene assembly）的计算能力[22, 50]。

纤毛虫是一种单细胞原生物。它们的名称得自它们表面的纤毛。它们包含两种细胞核：一个活跃的、包含功能基因的大细胞核（macronucleus），和一个功能上不活跃的小细胞核（micronucleus）。这个小细胞核只是用来进行遗传信息的交换。在交配的过程中，当两个纤毛虫交换完遗传信息、并且形成新的小细胞核后，它们必须实时地构成它们生存所必要的大细胞核。这个过程包括：重新排序（交换位置和反转）在小细胞核中的一些DNA片段，以及删除其它片段。这个从小细胞核得到大细胞核的过程叫做**基因重组**（gene rearrangement）[66]（如图7）。基因重组通过删除非编码序列，以及重新排列编码序列来得到正确的排序。多种被删除的非编码序列的功能还是未知的，它们占据了在小细胞核中的序列的很大一部分（达到98%）。作为一个例子，*Oxytricha nova*中的小细胞核基因*Actin I*是由9个编码片段组成的。这9个编码片段的顺序是3-4-6-5-7-9-2-1-8，并且，它们被8个非编码片段隔开。基因重组（也叫做基因整理（gene unscrambling））的指令们显然地被携带在小细胞

核的DNA中：指针序列出现在编码和非编码序列、以及一些RNA“模板”的连接部分。它们使有功能的大细胞核的基因重组成为可能。

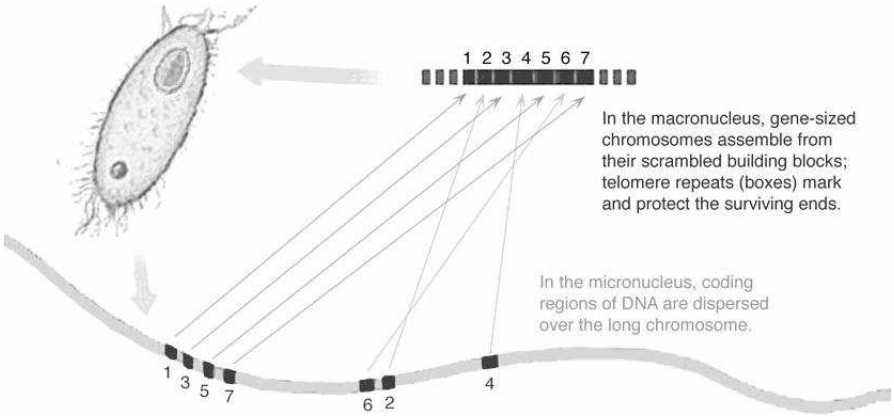

Fig. 7. 在鞭毛虫中，基因重组的概况。在大细胞核形成的过程中，小细胞核的分散的编码片段1-7（下面的）重组产生有功能的基因（上面的）。（原图来自Landweber, L. and Kari, L., The evolution of cellular computing: Nature's solution to a computational problem, *Proceedings of the DNA 4*, University of Pennsylvania, PA （取自L. Kari, H. Rubin, and D. Wood, Eds.）; *Biosystems*, 52, 3, 1999.）图中上部分文字：在大细胞核中，基因大小的染色体由杂乱的构建模块组成；重复的端粒（方块）标识并且保护残存的末端。图中下部分的文字：在小细胞核中，DNA中的编码区域被分散在长的染色体上。

从生物和计算的角度，基因重组的过程都是很吸引人的。从计算的角度，这方面的研究引出了很多全新的并且有挑战性的研究题目[22]。多种基因重组的模型都完全具有图灵机的计算能力，这个结果是很多研究结果之一[50]。从生物学的角度，计算机科学家和生物学家通过共同努力，得出了一个可信的（已经被一些实验数据支持）对于“生物件”（“bioware”）的假设。这些生物件实现了基因重组的过程。这个假设是根据模板指导的重组（template-guided recombination）[3, 63, 67]这个新的概念。

其它实现细胞计算的方法包括：在大肠杆菌内开发一个在细胞中的（*in vivo*）可编程的自动的有限自动机[59]；以及，利用细胞中存在的生化过程来设计和构造在细胞中的细胞逻辑门和遗传电路[49, 92, 93]。

7 DNA自我组装

DNA计算一个重要的成就是它对纳米科学的重要贡献。它对一些基础的问题提供了基于运算的见解。也许，其中最显著的是它对理解自我组装（self-assembly）的贡献。自我组装是纳米科学中一个关键的概念（见[70]，这是一个为计算机科学家写的关于自我组装的概述）。自我组装来自一个二维的tiling模型。这个模型是由Wang[90]定义和研究的。一个tile是一个四边带有颜色的正方形。把一些tile放在一个平面上时，它们形成一个整体的tiling的条件是：任意两个相邻的边具有相同的颜色。tiling系统可以被用来模拟一个图灵机的计算（例如，见[13, 90]）。通过使用自我组装的“DNA tiles”，Winfree等[94, 95]首次实现了计算性的tiling这个概念。本节描述了几个自我组装的模型以及自我组装的应用。这些应用包括：计算上的应用（7.1节），产生静态的DNA形状和图案（7.2节），设计动态的DNA纳米机械（7.3节）。

7.1 用来计算的DNA自我组装

分子的自我组装是一个自动的过程。在这个过程中，在没有外界的引导和干预的情况下，分子组合成一个由共价键连接的稳定的结构。这个结构是精确定义的。自我组装分为两类：分子内的和分子间的。通常所说的自我组装指的是分子间的自我组装，而分子内的自我组装叫做折叠（folding）。

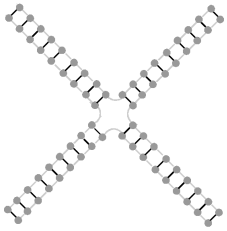

Fig. 8. DNA假日节点。根据设计，四个单链DNA片段形成一个有分支的结构。其中，每一个单链片段会参与在两个相邻的双链分支中。

DNA有很多坚硬的带有良好性质的结构。这些结构不是线性的双螺旋结构。这是在自我组装中使用DNA的重要的原因之一。例如，我们可以建造一种叫做假日节点（Holiday junction）的DNA结构[77]。其中，四个单链DNA片段自我组装，形成一个带有分支的结构（见图8）。在这个结构的设

计中，四个单链DNA片段都是部分地互补于另一个片段。由此，每一个片段会参与在两个相邻的双链分支中。

这类复杂的由自我组装形成的DNA结构已经被Winfree所使用。他介绍了一个用于DNA计算的二维的自我组装模型[95]。在这个模型中，构造矩形的DNA结构的技术类似于早期描述过的技术（例如，在[27]中演示的）。这些矩形结构的四角上带有单链的黏端。这些DNA结构就像tile一样，并且，可以根据设计进行二维的自我组装。根据它们的构成，一个tile的黏端们（单链DNA片段）只和其它tile上与它们Watson-Crick互补的黏端结合。结果是：这些tile自发地自我组装，形成一些平面的结构。这些平面结构可以根据设计进行确定性的或者非确定性的计算。

Yokomori等[101]介绍了另一个图灵通用（Turing-universal）的自我组装的模型。其它一些计算由复杂的DNA纳米结构的自我组装实现。这些计算包括：逐位累积的XOR[56]，以及二进制的计数器[5]。作为另一个例子，Seelig等[76]报告了基于DNA的数字逻辑电路的设计和实验性的实现。作者们演示了与（AND）、或（OR）、非门（NOT gates），以及信号复位（signal restoration）、放大（amplification）、反馈（feedback）、和级联（cascading）。

对于分子内的自我组装，Hagiya等[31]和Sakamoto等[74]提出了一个新的DNA计算的方法。这个方法包括一个自我操作（self-acting）的DNA分子。这个DNA分子同时包含输入、程序、和存储器。这个叫作Whiplash PCR的运算方式如下（见图9）。在5'端上，这个单链DNA片段包含一个$A \to B$类型的状态转换（state transition）。每个状态转换被编码为一个DNA的规则块（rule block），“$\overleftarrow{B}$-$\overleftarrow{A}$-*stopper sequence*”。在任何时刻，这个DNA片段的3'端包含“当前状态”(“current state”)的编码，例如，状态A。在第一步中，对于溶液的冷却会使代表状态A的3'端与它在规则块中的互补片段结合，也就是$\overleftarrow{A}$。在第二步中，根据下一个状态B的编码，结合中的A端被PCR延长。这个延长的过程被结束序列终止。在第三步中，通过提高温度，新的状态B脱离下来。之后，我们可以开始下一个状态转换。

使用这个模型的主要目的是，在一个单试管的反应中，所有的自我操作的分子可以进行并行运算。这使得多程序多输入的体系结构成为可能。Hagiya等[31]演示了如何使用这个模型从理论上学习μ-公式（μ-formulas）。随后，Winfree[96]演示了如何使用whiplash PCR，在$O(1)$生物步骤（biosteps）中，解决一些NP-完全问题。

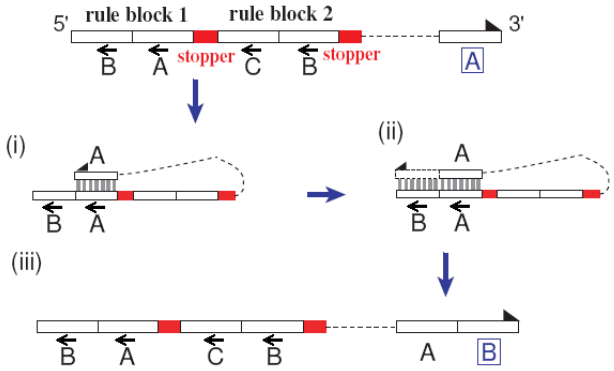

Fig. 9. Whiplash PCR （$\overleftarrow{A}$表示A的Watson-Crick互补片段）。在这个DNA片段的5'端上，规则块"$\overleftarrow{B}$-$\overleftarrow{A}$-stopper sequence"表示$A \rightarrow B$类型的状态转换。这个DNA片段的3'端包含当前状态A的编码。在（i）中，对于溶液的冷却使代表状态A的3'端与它在规则块中的互补片段结合。在（ii）中，结合中的A端被PCR延长，并且产生新的状态B。这个延长的过程被结束序列终止。在（iii）中，通过提高温度，新的状态B脱离下来。之后，我们可以开始下一个状态转换。

7.2 DNA纳米级形状

DNA纳米技术（见[78]，这是一个全面的概述）利用DNA独特的分子识别能力来创造复杂的结构，例如，带有有用属性的、由自我组装产生的、带有分支的DNA集合体。

由于DNA把自我组装的属性和可编程性结合起来了，所以，DNA是一种理想的纳米结构的建造模块。在纳米技术中，DNA被用作结构性的材料，而不是生物信息的携带者。通过使用DNA，令人印象深刻的纳米结构已经被创建。这包括二维的和三维的结构，例如，Sierpinski三角形[73]和立方体[17, 78]。

另一个设计主题在[81]中被介绍。作者们报告了一个单链DNA片段（长度为1669个核苷酸）被折叠成一个非常坚固的纳米级八面体结构。它的直径大约是22纳米。通过设计，这个长的DNA片段包含很多互补的区域。这些区域使得这个片段折叠，并且产生这个坚固的八面体。为了把这个DNA折叠为八面体，我们仅仅需要对溶液进行加热和冷却。所使用的溶液包含DNA、镁离子、和5个长度为40-mer的辅助性的DNA序列。此外，这个设计是首个可以克隆的三维DNA纳米结构。

在[72]中，Rothemund提出了一个简单的模型。这个模型使用一些短的DNA片段来引导一个长的单链DNA分子折叠为一个想要的形状。他演示了这个叫作**支架DNA折纸**（“scaffolded DNA origami”）方法的通用型。这个方法使得建造任何直径约为100纳米的DNA形状成为可能。这个方法所使用的技术与设计假日节点（7节）中所使用的技术相似。在假日节点的设计中，设计好的DNA片段组成复杂的结构。这使得一个单链DNA片段可能加入两个DNA螺旋结构。一个单链DNA片段先缠绕在一个螺旋结构中，然后再转换到另一个螺旋结构中。

DNA折纸的设计包含一些步骤。在第一步中，我们近似地建造一个想要得到的形状的几何模型。这个形状被近似地分割为一些圆柱形。这些圆柱形是一些DNA双螺旋结构的模型。这些DNA双螺旋结构最终被用来建造这个形状。在第二步中，我们通过来回地折叠一个长的单链“支架片段”（“scaffold strand”)来填满这个形状。折叠的形式像栅栏一样。在任何时刻，这个支架片段表示一个双螺旋的主片段（“main” strand）或者互补片段（“complement” strand)。（这个过程就像是使用铅笔画一条不断的线来填满一个形状。）当这个几何模型和折叠的路线设计好后，在第三步中，我们使用一个计算机程序来设计一组“钉书钉片段”（“staple strand”)。这些钉书钉片段提供了支架的Watson-Crick互补片段。通过设计，这些钉书钉片段与支架片段的一些部分结合，锁定支架片段所构成的形状。之后，通过仔细地调整这些钉书钉片段来减少结构中的拉力，以及优化结合的专一性和能量。

为了检测这个方法，Rothemund[72]使用病毒*M13mp18*中的环形的基因组DNA （长度为7249个核苷酸）作为一个支架。大约250种短的钉书钉片段被合成，并且与支架混合。每个短片段的数量是支架数量的100倍。这些片段在2小时内结合，并且，原子力显微镜（AFM）图片显示了，想要得到的形状确实形成了。通过合成6种不同的形状，这个方法的通用性被证明了。这6种形状包括：正方形、三角形、五角星、以及笑脸形状（图10)。

Fig. 10. （a）构造一个“DNA笑脸”的折叠路径。偶数个双螺旋结构被填充在这个形状里。(b)，(c）一个长的单链DNA片段自我组装，形成想要得到的形状（原子力显微镜图片)。（取自Rothemund, P., Nature, 440, 297, 2006.）

这个方法不仅提供了一种途径来获得那些与想要得到的形状相似的结构，并且允许我们创造一些带有任何形状的空洞或者表面图案的结构。此外，支架DNA折纸技术也许可以被用来设计三维的结构。

7.3 DNA纳米机械

除了静态的纳米级形状和图案，一组给人深刻印象的DNA纳米机械（见全面的概述[6, 53]）被设计出来了。这些DNA纳米机械在纳米制造、工程、和计算方面有潜在的用途。DNA纳米机械也就是基于DNA的纳米设备。它们把静态DNA结构转变为可以移动或者改变形态的器械。在1998年，一个DNA纳米机械被报告[100]。它是最早的DNA纳米机械之一。在此之后，DNA纳米机械得到了快速的发展。它们包括：可以在两种形态之间转换的分子开关[52]，DNA镊子[103]，在一个轨道上移动的DNA步行器[80, 82]，以及自发的分子马达[7, 30, 69]。

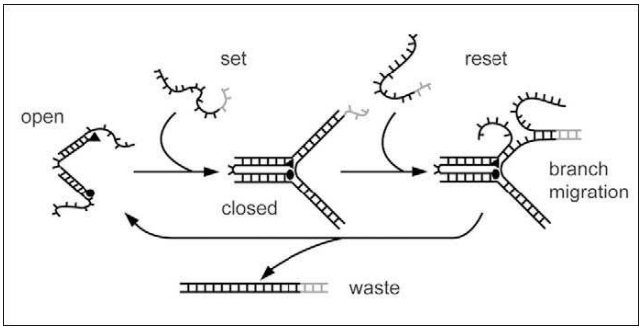

Fig. 11. “DNA镊子”。它的操作是通过重复地使用“设置”和“重置”片段。在张开的状态时，DNA镊子包括两个双螺旋状态的手臂（每个手臂带有一个短的单链尾端）。这两个手臂由一个短的单链合叶连接。手臂的单链尾端与设置片段杂交结合。这使得镊子关闭。分支移动（branch migration）可以移除设置片段。在分支移动的过程中，与设置片段完全互补的“重置”片段可以把设置片段从镊子手臂的尾端剥离下来。这使得镊子进入张开状态。（取自Liedl, T. et al., Nanotoday, 2, 36, 2007.）

我们使用一个叫作**分子镊子**（“molecular tweezer”）[103]的构造方法来阐明一些构造上的原理。DNA镊子包含两个手臂。他们的一部分和一个作为合叶的短的单链DNA片段形成不完全的双链结构（图11）。他们形成的结构像是一个张开的镊子。一个特殊设计的片段和这个镊子的两个手臂末端的

单链DNA片段互补。这个片段被叫作设定片段（“set strand”）。如果把这个设定片段加入到溶液中，它会与镊子的两个手臂的末端结合，使这个镊子进入“闭合”状态。即使在这个设定片段与两个手臂结合后，它的一个短的区域仍然保持单链状态。这个短的区域是一个起始点（toehold）。从这个起始点开始，通过与设定片段结合，一个新的重置片段（“reset strand”）从两个手臂上剥离设置片段。由此，这个镊子回到“张开”状态。这个镊子的一些变体也已经被报告了[21, 26, 58, 84]。

很多人致力于“步行器”（walking devices）的研究。一些可以沿着一个轨道行走的分子机械在[65, 79, 80, 82, 87]被提出。Shin和Pierce在[82]中介绍了一个有两个不同的脚的DNA设备。这个设备可以在一个直线的DNA轨道上按照一定的方向“行走”。这个轨道上有4种不同的单链片段。它们作为支撑点，并且周期性地出现在轨道上。这个步行器是一个双链DNA片段，并且带有两个单链状态的延长部分。这两个单链的延长部分是这个步行器的“腿”。特定的连接片段使步行器的两个腿和双链轨道上周期性出现的单链支撑点结合。行走的每一步需要按照顺序添加两个片段。在第一步中，通过使用片段置换，后面的一只脚被从轨道上抬起。片段置换是这样一个过程：由于可以产生一个更稳定的结构，一个单链的入侵片段可以替换一个双链片段中的一个组成片段。在第二步中，被抬起的脚被固定在另一只脚前面的位置上。布朗运动保证了步行器的移动性。添加“指令”片段的顺序保证了移动的方向性。根据尺蠖的移动模式，Sherman和Seeman开发了一个相似的步行器[80]: 这个步行器的前脚向前移动，后脚跟随前脚移动。

前面描述的步行器模型需要“燃料DNA”（fuel DNA），而且不能自发地行走。最近，[79]提出了一个有三条腿的分子步行器。这个步行器可以自发地在一个设计好的二维的或者三维的路线上行走。这个步行器使用酶作为动力的来源。它的轨道是由很多单链DNA支撑点组成。这些支撑点构成一个路径。

在本节的最后，我们介绍关于DNA纳米设备与细胞的遗传机制结合的研究。分子有机体展现了一些由基因、RNA、和蛋白质组成的，复杂的生物化学反应网络。这些反应网络包括：基因的激活与抑制，从基因到RNA的翻译，从RNA到蛋白质的翻译。通过修改这些遗传机制，我们可以设计全新的在试管中的（*in vitro*），以及潜在的、在细胞中的（*in vivo*）分子机械[53]。一些使用遗传机制来控制DNA纳米设备的尝试已经被完成[47, 48, 61, 62]。例如，Noireaux、BarZiv、和Libchaber在他们的创新性的工作中[61, 62] 演示了无细胞遗传电路组装（cell-free genetic circuit assembly）的一些原理。其中，

“无细胞”指的是实验发生在试管里，而不是细胞里。总的来说，DNA纳米设备与遗传机制的成功结合引出了一些在生物科技、生物工程、和医学方面有希望的可能的应用。

8 总结

1994年的第一个DNA计算的实验令人兴奋的主要原因是，DNA计算给我们提供了一个全新地看待和进行计算的方式：通过使用酶来剪切和粘贴DNA片段，通过使用DNA片段的Watson-Crick互补性，选择包含一个特定模式的DNA片段，等等。这种看待计算的新方式有潜力改变“计算”这个词的含义，并且，DNA计算已经对计算机科学作出了很大的贡献。与此同时，对于细胞有机体的计算能力的研究有潜力揭示生物信息的原则，并且使我们能够利用细胞的计算能力。

本章的主要目的是通过以下方面做一个DNA计算的浅析：理论的和实验的方面，经典的和最新的趋势，分子计算在计算理论和计算科技上的影响，以及对于一些生物过程的基本机制的理解。

DNA计算是一个快速发展的领域，并且在最近几年出现了一些有趣的新发展。实验上的进展包括：圆柱体和莫比乌斯带（Möbius strip）的自我组装[29]，使用光做燃料的新的纳米机械的设计[51]。DNA计算的理论方面的进展包括：无伪结的（pseudoknot-free）DNA/RNA词[45]，时间上最优化的三维形状的自我组装[9]，为了合成大规模电路而提出的简单的可扩展的DNA逻辑门[68]。

在试管中的、和在细胞中的DNA计算的研究仅仅是一个展望的第一步。它们最终会使分子计算成为计算的一个切实可行的辅助工具，并且，更深刻地认识在活的有机体中，生物过程的计算的本质。

9 致谢

本研究项目是由加拿大自然科学与工程研究会（NSERC）的探索基金（Discovery Grant），和Lila Kari的加拿大研究主席奖金（Canada Research Chair Award）支持的。我们感谢Bo Cui，Elena Czeizler，Eugen Czeizler，Zachary Kincaid，Shinnosuke Seki，以及匿名的审阅人，他们为这篇文章提供了宝贵的建议和意见。我们还感谢Yan Zeng，她为这篇文章提供了图9和8。

References

1. L.Adleman, *Molecular computation of solutions to combinatorial problems*, Science 266 (1994) 583-585.
2. M.Amos, *Theoretical and Experimental DNA Computation*, Springer Verlag, Natural Computing Series 2005.
3. A.Angeleska, N.Jonoska, M.Saito, L.Landweber, *RNA-guided DNA assembly*, Journal of Theoretical Biology 248 (2007) 706-720.
4. C.Bancroft, T.Bowler, B.Bloom, C.Clelland, *Long-term storage of information in DNA*, Science 293 (2001) 1763-1765.
5. R.Barish, P.Rothemund, E.Winfree, *Two computational primitives for algorithmic self-assembly: copying and counting*, Nanoletters 5 (2005) 2586-2592.
6. J.Bath, A.Turberfield, *DNA nanomachines*, Nature Nanotechnology 2 (2007) 275-284.
7. J.Bath, S.Green, A.Turberfield, *A free-running DNA motor powered by a nicking enzyme*, Angewandte Chemie International Edition 44 (2005) 4358-4361.
8. E.Baum, *How to build an associative memory vastly larger than the brain*, Science 268 (1995) 583-585.
9. F.Becker, E.Remila, N.Schabanel, *Time optimal self-assembling of 2D and 3D shapes: the case of squares and cubes*, Prelim. Proc. of DNA 14[3], Prague, Czech Republic, (A.Goel, F.Simmel, P.Sosik, Eds.) June (2008) 78-87.
10. Y.Benenson, T.Paz-Elizur, R.Adar, E.Keinan, Z.Livneh, E.Shapiro, *Programmable and autonomous computing machine made of biomolecules*, Nature 414 (2001) 430-434.
11. Y.Benenson, R.Adar, T.Paz-Elizur, Z.Livneh, E.Shapiro, *DNA molecule provides a computing machine with both data and fuel*, PNAS 100 USA (2003) 2191-2196.
12. Y.Benenson, B.Gil, U.Ben-Dor, R.Adar, E.Shapiro, *An autonomous molecular computer for logical control of gene expression*, Nature 429 (2004) 423-429.
13. R. Berger, *Undecidability of the domino problem*, Memoirs of the Amer. Math. Soc. 66 (1966), 72pp.
14. J.Berstel, D.Perrin, *Theory of Codes*, Academic Press Inc. Orlando Florida 1985.
15. R.Braich, N.Chelyapov, C.Johnson, P.Rothemund, L.Adleman, *Solution of a 20-variable 3-SAT problem on a DNA computer*, Science 296 (2002) 499-502.
16. J.Chen, R.Deaton, Y.Wang, *A DNA-based memory with in vitro learning and associative recall*, Natural Computing 4 (2005) 83-101.

[3] *DNA n* denotes the main international conference in the field of DNA Computing, *The nth International Conference on DNA Computing.*

17. J.Chen, N.Seeman, *Synthesis from DNA of a molecule with the connectivity of a cube*, Nature 350 (1991) 631-633.
18. J.Cox, *Long-term data storage in DNA*, Trends in Biotechnology 19 (2001) 247-250.
19. E. Czeizler, L. Kari, S. Seki, *On a special class of primitive words*, Proc. of Mathematical Foundations of Computer Science (MFCS), LNCS 5162 (2008) 265-277.
20. M.Daley, L.Kari, *DNA computing: Models and implementations*, Comments on Theoretical Biology 7 (2002) 177-198.
21. B.Ding, N.Seeman, *Operation of a DNA robot arm inserted into a 2D DNA crystalline substrate*, Science 314 (2006) 1583-1585.
22. A.Ehrenfeucht, T.Harju, I.Petre, D.Prescott, G.Rozenberg, *Computation in Living Cells: Gene Assembly in Ciliates*, Springer Verlag, Natural Computing Series 2004.
23. D.Faulhammer, A.Cukras, R.Lipton, L.Landweber. *Molecular computation: RNA solutions to chess problems*, PNAS 97 (2000), 1385-1389.
24. U.Feldkamp, W.Banzhaf, H.Rauhe, *A DNA sequence compiler*, Prelim. Proc. of DNA 6 Leiden, Netherlands, (A.Condon, G.Rozenberg, Eds.) June (2000).
25. U.Feldkamp, S.Saghafi, W.Banzhaf, H.Rauhe, *DNA sequence generator: A program for the construction of DNA sequences*, Proc. of DNA 7, (N.Jonoska, N.Seeman, Eds.) LNCS 2340 (2002) 23-32.
26. L.Feng, S.Park, J.Reif, H.Yan, *A two state DNA lattice switched by DNA nanoactuator*, Angewandte Chemie International Edition 42 (2003) 4342-4346.
27. T.Fu, N.Seeman, *DNA double cross-over structures*, Biochemistry 32 (1993) 3211-3220.
28. M.Garzon, C.Oehmen, *Biomolecular computation in virtual test tubes*, Proc. of DNA 7, (N.Jonoska, N.Seeman, Eds.) LNCS 2340 (2002) 117-128.
29. M.Gopalkrishnan, N.Gopalkrishnan, L.Adleman, *Self-assembly of cylinders and Möbius strips by DNA origami*, Prelim. Proc. of DNA 14, Prague, Czech Republic, (A.Goel, F.Simmel, P.Sosik, Eds.) June (2008) 181.
30. S.Green, D.Lubrich, A.Turberfield, *DNA hairpins: fuel for autonomous DNA devices*, Biophysical Journal 91 (2006) 2966-2975.
31. M.Hagiya, M.Arita, D.Kiga, K.Sakamoto, S.Yokoyama, *Towards parallel evaluation and learning of boolean μ-formulas with molecules*, Proc. of DNA 3 (H.Rubin, D.Wood, Eds.) DIMACS 48 (1997) 57-72.
32. A.Hartemink, D.Gifford, *Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation*, Proc. of DNA 3 (H.Rubin, D.Wood, Eds.) DIMACS 48 (1997) 25-38.

33. A.Hartemink, D.Gifford, J.Khodor, *Automated constraint-based nucleotide sequence selection for DNA computation*, Proc. of DNA 4 (L.Kari, H.Rubin, D.Wood, Eds.) Biosystems 52 (1999) 227-235.
34. J.Hopcroft, J.Ullman, R.Motwani, *Introduction to Automata Theory, Languages and Computation*, 2nd edition, Boston: Addison Wesley 2001.
35. S.Hussini, L.Kari, S.Konstantinidis, *Coding properties of DNA languages*, Theoretical Computer Science 290 (2003) 1557-1579.
36. N.Jonoska, K.Mahalingam, J.Chen, *Involution codes: with application to DNA coded languages*, Natural Computing 4 (2005) 141-162.
37. N.Jonoska, L.Kari, K.Mahalingam, *Involution solid and join codes*, Intl. Conf. on Developments in Language Theory (DLT), LNCS 4036 (2006) 192-202.
38. L.Kari, *DNA Computing: arrival of biological mathematics*, The Mathematical Intelligencer 19 (1997) 9-22.
39. L.Kari, R.Kitto, G.Gloor, *A computer scientist's guide to molecular biology*, Soft Computing, (G.Paun, T.Yokomori, Eds.), 5 (2001) 95-101.
40. L.Kari, S.Konstantinidis, E.Losseva, G.Wozniak, *Sticky-free and overhang-free DNA languages*, Acta Informatica 40 (2003) 119-157.
41. L.Kari, S.Konstantinidis, E.Losseva, P.Sosik, G.Thierrin, *Hairpin structures in DNA words*, Proc. of DNA 11, (A.Carbone, N.Pierce, Eds.) LNCS 3892 (2006) 158-170.
42. L.Kari, K.Mahalingam, *Watson-Crick conjugate and commutative words*, DNA 13, (M.Garzon, H.Yan, Eds.) LNCS 4848 (2008) 273-283.
43. L.Kari, K.Mahalingam, *Watson-Crick palindromes in DNA computing*, Natural Computing (2008) to appear.
44. L.Kari, S.Konstantinidis, P.Sosik, *Bond-free languages: formalizations, maximality and construction methods*, International Journal of Foundations of Computer Science 16 (2005) 1039-1070.
45. L. Kari, S. Seki, *On pseudoknot-bordered words and their properties*, Journal of Computer and System Sciences, (2008), doi:10.1016/j.jcss.2008.08.002.
46. S.Kashiwamura, M.Yamamoto, A.Kameda, T.Shiba, A.Ohuchi, *Potential for enlarging DNA memory: the validity of experimental operations of scaled-up nested primer molecular memory*, Biosystems 80 (2005) 99-112.
47. J.Kim, J.Hopfield, E.Winfree, *Neural network computation by in vitro transcriptional circuits*, Advances in Neural Information Processing Systems (NIPS) 17 (2004) 681-688.
48. J.Kim, K.White, E.Winfree, *Construction of an in vitro bistable circuit from synthetic transcriptional switches*, Molecular Systems Biology 2 (2006) doi:10.1038/msb4100099.

49. T.Knight Jr., G.Sussman, *Cellular gate technology*, Unconventional Models of Computation (1998) 257-272.
50. L.Landweber, L.Kari, *The evolution of cellular computing: nature's solution to a computational problem*, Proc. of DNA 4 (L.Kari, H.Rubin, D.Wood, Eds.) Biosystems 52 (1999) 3-13.
51. X.Liang, H.Nishioka, N.Takenaka, H.Asanuma, *Construction of photon-fuelled DNA nanomachines by tethering azobenzenes as engines*, Prelim. Proc. of DNA 14, Prague, Czech Republic, (A.Goel, F.Simmel, P.Sosik, Eds.) June (2008) 17-25.
52. T.Liedl, M.Olapinski, F.Simmel, *A surface-bound DNA switch driven by a chemical oscillator*, Angewandte Chemie International Edition 45 (2006) 5007-5010.
53. T.Liedl, T.Sobey, F.Simmel, *DNA-based nanodevices*, Nanotoday 2 (2007) 36-41.
54. R.Lipton, *DNA solution of hard computational problems*, Science 268 (1995) 542-545.
55. M.Lothaire, *Combinatorics on Words*, Cambridge University Press, Cambridge UK 1997.
56. C.Mao, T.LaBean, J.Reif, N.Seeman, *Logical computation using algorithmic self-assembly of DNA triple-crossover molecules*, Nature 407 (2000) 493-496.
57. A.Marathe, A.Condon, R.Corn, *On combinatorial word design*, Journal of Computational Biology 8 (2001) 201-219.
58. J.Mitchell, B.Yurke, *DNA scissors*, Proc. of DNA 7, (N.Jonoska, N.Seeman, Eds.) LNCS 2340 (2002) 258-268.
59. H.Nakagawa, K.Sakamoto, Y.Sakakibara, *Development of an in vivo computer based on Escherichia Coli*, Proc. of DNA 11, (A.Carbone, N.Pierce, Eds.) LNCS 3892 (2006) 203-212.
60. A.Neel, M.Garzon, P.Penumatsa, *Improving the quality of semantic retrieval in DNA-based memories with learning*, International conference on Knowledge-Based Intelligent Information and Engineering Systems, LNCS 3213 (2004) 18-24.
61. V.Noireaux, A.Libchaber, *A vesicle bioreactor as a step toward an artificial cell assembly*, PNAS 101 (2004) 17669-17674.
62. V.Noireaux, R.Bar-Ziv, A.Libchaber, *Principles of cell-free genetic circuit assembly*, PNAS 100 (2003) 12672-12677.
63. M.Nowacki, V.Vijayan, Y.Zhou, K.Schotanus, T.Doak, L.Landweber, *RNA-mediated epigenetic programming of a genome-rearrangement pathway*, Nature 451 (2008) 153-158.
64. G.Paun, G.Rozenberg, A.Salomaa, *DNA Computing - New Computing Paradigms*, Springer-Verlag 1998.

65. R.Pei, S.Taylor, D.Stefanovic, S.Rudchenko, T.Mitchell, M.Stojanovic, *Behaviour of polycatalytic assemblies in a substrate displaying matrix*, Journal of American Chemical Society 128 (2006) 12693-12699.
66. D.Prescott, *The DNA of ciliated protozoa*, Microbiology and Molecular Biology Reviews 58 (1994) 233-267.
67. D.Prescott, A. Ehrenfeucht, G. Rozenberg, *Template- guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates*, Journal of Theoretical Biology 222 (2003) 323-330.
68. L.Qian, E.Winfree, *A simple DNA gate motif for synthesizing large-scale circuits*, Prelim. Proc. of DNA 14, Prague, Czech Republic, (A.Goel, F.Simmel, P.Sosik, Eds.) June (2008) 139-151.
69. J.Reif, *The design of autonomous DNA nanomechanical devices: Walking and rolling DNA*, Proc. of DNA 8, (M.Hagiya, A.Ohuchi, Eds.) LNCS 2568 (2003) 22-37.
70. J.Reif, T.LaBean, *Autonomous programmable biomolecular devices using self-assembled DNA nanostructures*, Communications of the ACM 50 (2007) 46-53.
71. J.Reif, T.LaBean, M.Pirrung, V.Rana, B.Guo, C.Kingsford, G.Wickham, *Experimental construction of very large scale DNA databases with associative search capability*, Proc. of DNA 7, (N.Jonoska, N.Seeman, Eds.) LNCS 2340 (2002) 231-247.
72. P.Rothemund, *Folding DNA to create nanoscale shapes and patterns*, Nature 440 (2006) 297-302.
73. P.Rothemund, N.Papadakis, E.Winfree, *Algorithmic self-assembly of DNA Sierpinski triangles*, PLoS Biology 2 (2004) 2041-2053.
74. K.Sakamoto, H.Gouzu, K.Komiya, D.Kiga, S.Yokoyama, T.Yokomori, M.Hagiya, *Molecular computation by DNA hairpin formation*, Science 288 (2000) 1223-1226.
75. A.Salomaa, *Formal Languages*, Academic Press 1973.
76. G.Seelig, D.Soloveichik, D.Zhang, E.Winfree, *Enzyme-free nucleic acid logic circuits*, Science 314 (2006) 1585-1588.
77. N.Seeman, *DNA in a material world*, Nature 421 (2003) 427-431.
78. N.Seeman, *Nanotechnology and the double-helix*, Scientific American 290 (2004) 64-75.
79. H.Sekiguchi, K.Komiya, D.Kiga, M.Yamamura, *A realization of DNA molecular machine that walks autonomously by using a restriction enzyme*, Proc. of DNA 13, (M.Garzon, H.Yan, Eds.) LNCS 4848 (2008) 34-65.
80. W.Sherman, N.Seeman, *A precisely controlled DNA biped walking device*, Nano Letters 4 (2004) 1203-1207.

81. W.Shih, J.Quispe, G.Joyce, *A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron*, Nature 427 (2004) 618-621.
82. J.Shin, N.Pierce, *A synthetic DNA walker for molecular transport*, Journal of American Chemical Society 126 (2004) 10834-10835.
83. H.J.Shyr, *Free Monoids and Languages*, Hon Min Book Company 2001.
84. F.Simmel, B.Yurke, *Using DNA to construct and power a nanoactuator* Physical Review E63 (2001) 041913.
85. G.Smith, C.Fiddes, J.Hawkins, J.Cox, *Some possible codes for encrypting data in DNA*, Biotechnology Letters 25 (2003) 1125-1130.
86. D.Soloveichik, E.Winfree, *The computational power of Benenson automata*, Theoretical Computer Science 344 (2005) 279-297.
87. Y.Tian, Y.He, Y.Chen, P.Yin, C.Mao, *A DNAzyme that walks processively and autonomously along a one-dimensional track*, Angewandte Chemie International Edition 44 (2005) 4355-4358.
88. Y.Tsuboi, Z.Ibrahim, O.Ono, *DNA-based semantic memory with linear strands*, International Journal of Innovative Computing, Information and Control 1 (2005) 755-766.
89. D.Tulpan, H.Hoos, A.Condon, *Stochastic local search algorithms for DNA word design*, Proc. of DNA 8, (M.Hagiya, A.Ohuchi, Eds.) LNCS 2568 (2003) 229-241.
90. H.Wang, *Proving theorems by pattern recognition - II*, Bell System Tech. J. 40 (1961) 1-41.
91. J.Watson, N.Hopkins, J.Roberts, J.Steitz, A.Weiner, *Molecular Biology of the Gene*, Menlo Park, CA: Benjamin Cummings 1987.
92. R.Weiss, G.Homsy, T.Knight, Jr., *Toward in-vivo digital circuits*, Evolution as Computation, Natural Computing Series (2002) 275-295.
93. R. Weiss, S. Basu, *The device physics of cellular logic gates*, Proc. of The First Workshop on Non-Silicon Computation (2002) 54-61.
94. E. Winfree, F. Liu, L. Wenzler, N.Seeman, *Design and Self-Assembly of Two-Dimensional DNA Crystals*, Nature 394 (1998) 539-544.
95. E.Winfree, X.Yang, N.Seeman, *Universal computation via self-assembly of DNA: some theory and experiments*, Proc. of DNA 2 (L.Landweber, E.Baum, Eds.) DIMACS 44 (1996) 191-213.
96. E.Winfree, *Whiplash PCR for $O(1)$ computing*, Caltech Technical Report, CaltechCSTR:1998.23 (1998).
97. P.Wong, K.Wong, H.Foote, *Organic data memory using the DNA approach*, Communications of the ACM 46 (2003) 95-98.
98. N.Yachie, K.Sekiyama, J.Sugahara, Y.Ohashi, M.Tomita, *Alignment based approach for durable data storage into living organisms*, Biotechnology Progress 23 (2007) 501-505.

99. M.Yamamoto, S.Kashiwamura, A.Ohuchi, *DNA memory with 16.8M addresses*, Proc. of DNA 13, (M.Garzon, H.Yan, Eds.) LNCS 4848 (2008) 99-108.
100. X.Yang, A.Vologodskii, B.Liu, B.Kemper, N.Seeman, *Torsional Control of Double Stranded DNA Branch Migration*, Biopolymers 45 (1998) 69-83.
101. T.Yokomori, *Yet another computation model of self-assembly*, Proc. of DNA 5 (E.Winfree, D.Gifford, Eds.) DIMACS Series 54 (1999) 153-167.
102. S.S.Yu, *Languages and Codes*, Lecture Notes, Department of Computer Science, National Chung-Hsing University, Taichung, Taiwan 2005.
103. B.Yurke, A.Turberfield, A.Mills, F.Simmel, J.Neumann, *A DNA-fuelled molecular machine made of DNA*, Nature 406 (2000) 605-608.