

# DNA 分子计算与 DNA 计算机的研究进展

俞洋, 缪准扣\*, 宋世平, 樊春海\*

上海大学计算机工程与科学学院, 上海 200072;

中国科学院上海应用物理研究所, 上海 201800

\* 联系人, E-mail: [fchh@sinap.ac.cn](mailto:fchh@sinap.ac.cn); [hkmiao@shu.edu.cn](mailto:hkmiao@shu.edu.cn)

2007-11-07 收稿, 2007-12-18 接受

国家自然科学基金(批准号: 20725516, 60673115)、国家重点基础研究发展计划(批准号: 2007CB3108001)和上海市科委(批准号: 03DZ14025, 0752nm021)资助项目

**摘要** 生物分子计算与 DNA 计算机是计算机科学和分子生物学交叉产生的新兴领域. DNA 计算机的特点是具有超强的并行运算能力和巨大的数据存储能力, 因而被认为有望解决电子计算机所面临的评价问题. 本文在介绍 DNA 计算机的基本概念基础上, 围绕 DNA 计算机的原理、计算模型和在多方面的应用等关键问题, 分析讨论了粘贴模型、剪接模型和等价检查模型等常用的 DNA 计算模型, 并对 DNA 计算机在 NP 问题、遗传分析与临床诊治、防伪和译码技术以及游戏与机器人等领域的研究进展和应用前景进行了探讨. 最后讨论了 DNA 计算机未来可能的发展方向.

**关键词**

分子计算  
DNA 计算机  
DNA 结构  
分子逻辑门  
分子医生

电子计算机的诞生是人类历史上一个里程碑式的突破, 可以说在很大程度上改变了人类的生活方式. 由于集成电路技术的飞速进展, 电子计算机本身的发展也是惊人的, 其运算速度较其原始模型已提高了约 10 亿倍. 然而, 随着这种惊人的技术发展, 人们不断提出这样的问题, 摩尔定律是否还能继续有效? 电子计算机是否可能走入穷途末路? 目前普遍公认的现状是, 越来越高的集成度要求向传统的集成电路工艺提出了严峻的挑战, 而集成电路的发展已经越来越接近技术所能容许的极限, 首先是电子通道的布线因越来越密而变得越来越难, 因为晶体管之间的连接导线的厚度已被蚀刻到只有  $0.18 \mu\text{m}$ , 电路线宽在  $0.1 \mu\text{m}$  以下将不可避免地达到仅有单个分子大小的物理学极限; 其次, 在更小的硅芯片上布设更密的电路将导致过热的高温而使芯片难以承受. 多数观点认为, 传统电子计算机在信息存储和处理方面取得更大的发展难度很大. 由于新兴的高新技术研究对计算机的性能不断提出更高要求, 例如人工智能的实现取决于计算机电路的密度和复杂性, 因此目前以半导体技术为基础的电子电路难以产生真正的认知能力.

基于这样的认识, 人们意识到研制基于全新原理的计算机势在必行. 如果回顾历史的话, 我们发现电子计算机的发展经历了这样一个历程. 1936 年英国数学家图灵提出了一种通用编程计算机器的概念, 即著名的图灵机. 之后在 20 世纪 40 年代基于真空电子管的计算机才问世, 最终发展为今天基于硅的电子计算机. 然而, 很多自然界中的物理化学过程都蕴涵着类似的信息处理机制, 如果能抽象出对应的计算模型就有可能利用这些过程进行计算. 事实上, 细胞内的 DNA, RNA 等生物分子进行生化反应的过程和图灵机极其相似, 它们都是从一个字母表中抽出一串字符串, 并对这一特定信息进行处理, 同时根据给定的规则对字符串进行修改或增减, 从而一步步地沿着字符串运行. 基于这一认识, 人们已经开发了诸如生物计算、量子计算等新一代计算方法. 生物分子的大小在纳米尺度, 同时生物分子具有良好的可操作性与强大的识别和信号转导能力, 这就为研制分子层次的利用生物分子元件组装成的生物计算机提供了可能(图 1). 这一领域目前正吸引着越来越多的研究者投身其中, 其中 DNA 计算机的诞生和发展尤其引起了广泛的关注 [1-6].

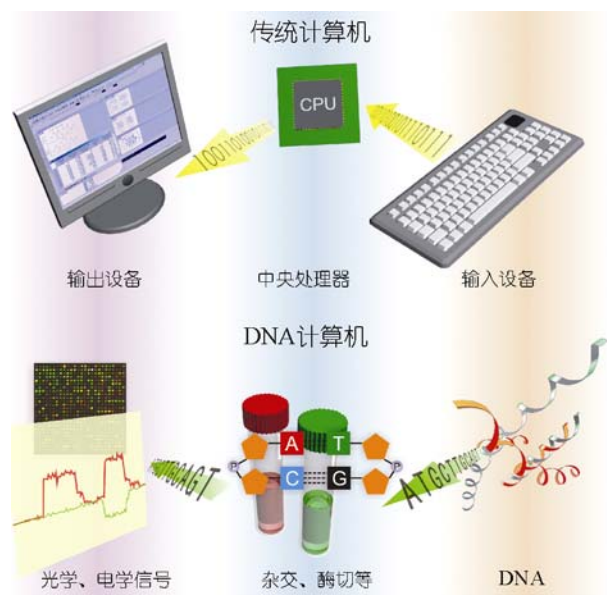


图1 传统的电子计算机与DNA计算机的示意图

两者的基本单元都是由输入设备(input)、中央处理器(CPU)和输出设备(output)构成。在电子计算机中,操作人员将待处理数据按照一定规则输入到CPU,经过CPU的运算再通过电信号将处理后的数据输出到输出设备上。类似的,DNA计算机是将信息编码在DNA链的碱基序列(AGTC)中,经过一系列的DNA杂交、酶切等生化反应后,以光学或电学信号的形式将最终计算结果输出

## 1 DNA 计算的原理

DNA(脱氧核糖核酸)是生物遗传的物质基础。1953年,Watson和Crick通过X射线衍射方法得出DNA的双螺旋结构。DNA双螺旋结构的骨架由脱氧核糖和磷酸基通过酯键交替连接而成,两条骨架绕一共同轴心以右手方向盘旋,相互平行而走向相反形成双螺旋结构。DNA由4种碱基:腺嘌呤、鸟嘌呤、胞嘧啶、胸腺嘧啶(分别简称为A、G、C和T)组合而成,碱基位于螺旋的内侧,它们以垂直于螺旋轴的取向通过糖苷键与骨架糖基相连。碱基的排列组合存储着生物遗传信息。DNA的一个重要特性是DNA链可以通过碱基互补配对作用形成杂合的双链,而且配对具有高度的特异性,即A只能与T,G只能与C结合。同一平面的碱基在两条骨架之间形成碱基对。G和C之间有3个氢键,而A和T之间只有两个氢键。DNA计算机即是通过这些特异性的配对作用而构建的,即将运算信息排列于DNA上,并通过特定DNA片段之间的相互作用来得出运算结果(图1)。事实上,DNA分子即可以看作4种不同的符号A、G、C、T组成的串,可以将字母集合 $\Sigma = \{A, G, C, T\}$ 按照碱

基互补配对原则进行编码。

利用DNA分子的特异性配对原理可以很方便地实现分子逻辑开关。美国Scripps研究所Ghadiri研究小组<sup>[2]</sup>在2003年提出了通过荧光标记的DNA分子构建分子逻辑门的方法。利用DNA分子之间的配对作用和多种荧光分子之间的Foster共振能量转移效应,他们构建了3种光学逻辑门:AND、NAND和INHIBIT。我们此前则设计了一种完全由DNA碱基构成的逻辑门<sup>[8]</sup>。在这个工作中选用了一种铜离子依赖的DNA核酶(一种具有酶活性的特殊DNA结构),构建YES门、NOT门等基本的逻辑门。基于这一方法也可以实现逻辑门之间的连接,形成较为复杂的逻辑门,如AND(A, NOT(B), NOT(C))等。

DNA计算机的主要特征之一就是实施大规模的并行运算。DNA计算之父Adleman指出<sup>[4]</sup>,DNA计算机一步可完成 $10^{20}$ 次运算,其运算速度大大超过电子计算机的运算速度;同时,生物计算机每消耗1J的能量,可以完成 $10^{19}$ 次运算,其能量损耗及能量效率远远优于电子计算机。并行运算可以解决电子计算机难以解决的计算问题,如非多项式问题(non-polynomial problem, NP问题)。NP问题的计算时间,随着变量数目的增加呈指数增加,因此电子计算机在变量数目较大时就对NP问题无能为力了。

哈密尔顿路径问题(Hamiltonian path problem, HPP问题)即是一类典型的NP问题。HPP问题是指推销员在一个具有 $n$ 个城市和 $m$ 条航线的地图(有向图)中,寻找从某一城市出发到达目的城市的一条通路,并且要求经过且仅经过其他所有城市一次。1994年,Adleman<sup>[4]</sup>利用DNA计算机实现了具有7个城市和13条航线的哈密尔顿路径问题。其解决方案分为3个部分:(1)先用长度为20个核苷酸的不同DNA序列,编码7个城市设为 $O_i (i = 0, 1, 2, \dots, 6)$ ,然后将13条 $i_j$ 边编码为 $O_i_j, O_i_j$ 的前10个核苷酸是 $O_i$  3'端的10个核苷酸,而后10个核苷酸为 $O_j$  5'端的10个核苷酸。在分子生物学反应中,将等摩尔量的 $O_i$ 的互补链和 $O_i_j$ 混合,以 $O_i$ 的互补链为模板,使对应的边进行反应,进而产生各种随机连接序列(路径)。(2)以 $O_0$ 序列和 $O_6$ 的互补序列为引物,通过PCR扩增随机序列,不满足条件的“路径”序列将不被扩增。(3)应用电泳分离扩增产物,获取目的双链DNA(dsDNA),此DNA序列即代表通过7个城市的“路径”序列。(4)上述双链DNA变性后与磁颗粒固定的 $O_1$ 互补链结合,选取通

过城市 1 的通路. 然后依次用磁颗粒固定的 $O_2$ ,  $O_3$ ,  $O_4$ ,  $O_5$ 的互补链重复操作, 获取同时通过所有城市的路径. (5) 将最终获取的DNA序列通过聚合酶链式反应(PCR)扩增, 用电泳分离输出结果.

通过这一模型实验, Adleman 成功展示了利用DNA 计算机解决 HPP 问题的可能性. 重要的是, 电子计算机进行 HPP 问题计算时其运算步骤与节点数目呈指数关系; 而 DNA 计算具有高度的并行性, 其运算步骤与城市节点的数目呈线性关系. 因此, DNA 计算在解决大变量数目的 NP 问题时具有常规电子计算机所不可比拟的优势.

由 Adleman 的例子也可看出, DNA 计算可分为 3 个过程. 第一是编码合成过程, 即对数学问题所有可能的解答, 设计出用不同的 DNA 序列进行编码, 并合成大量的各种 DNA 编码序列. 第二是清除运算过程, 通过 DNA 链的互补结合以及核酸酶的作用, 大量清除那些代表不正确答案的 DNA 序列, 而保留代表正确答案的 DNA 序列. 第三是结果读出过程, 即通过 PCR 技术将正确答案阅读出来.

## 2 DNA 计算的模型

DNA 计算模型的提出和发展对于DNA 计算机研究来说具有至关重要的作用. 经过十多年的发展, 研究者基于分子生物学操作的可能性已经提出了粘贴模型(sticker model)、剪接模型(splicing model)、等价检查模型(equality checking model)等具有计算完备性的模型. 这些模型都被证明可以和图灵机等价<sup>[9]</sup>. 然而在目前研究阶段, 已有的模型只能实现加法、乘法等运算, 而不能进行减法和除法运算. 然而可以预料通过计算机专家和生物学家们的密切合作, 真正实用的能够实现电子计算机功能的计算机模型必将可以实现. 以下将对几种常见模型进行简述.

### 2.1 粘贴模型

粘贴模型是按照DNA 碱基互补的特性进行配对识别操作的DNA 检索模型<sup>[10]</sup>, 即在一条长的DNA 单链上随机选取一些位点, 并设计相应的配对DNA 片段, 这样就形成了单、双链间隔的DNA. 然后通过合并、分离、设置和清除等设定操作即可实现DNA 计算. 这种方式的优势在于不需要酶的参与即可实现DNA 计算<sup>[10]</sup>. 然而其计算完备性还没有得到进一步研究. 1998 年, 研究者基于Adleman提出的概念建立了计算完备的粘贴模型<sup>[11,12]</sup>. 他们通过形式语言的

方法将粘贴系统表示为一个四元组 $\gamma = (V, \rho, A, D)$ , 其中 $V$ 是字母表,  $\rho$ 是 $V$ 上的互补关系,  $A$ 和 $D$ 都是有限子集. 通过粘贴运算, 最终可以得到完整的双链DNA 分子而实现DNA 计算. 到目前为止, 基于这一模式的粘贴系统产生的所有语言都是正则语言. 粘贴模型具有体系简单的优点, 在运算中不涉及到酶参与的DNA 延伸等复杂操作, 因而很有希望成为一个通用的计算模型.

### 2.2 剪接模型

剪接模型是另一类被证明具有计算完备性的系统. 该模型是将DNA 重组行为抽象为数学上的剪接操作, 即把DNA 链看作四字符集形成的串, 然后通过专门处理字符和字符串的形式语言理论来研究DNA 分子的重组行为. 需要指出的是, DNA 分子剪接是通过分子生物学操作的, 即利用限制性内切酶、DNA 连接酶、DNA 聚合酶和外切酶等实现DNA 的切割、连接与扩增. 其中限制性内切酶相当于分离算子, DNA 连接酶相当于连接算子, DNA 聚合酶相当于复制算子, 外切酶可以作为删除算子. 剪接系统是一个四元组 $r = (\Sigma, T, A, R)$ , 其中 $\Sigma$ 是一个字符集,  $T$ 是终极字符集,  $A$ 是 $\Sigma^*$ 上的多重集,  $R$ 是剪接规则的集合. Freund 等<sup>[13]</sup>认为基于剪接系统的DNA 计算可以发展成为通用DNA 计算机. 事实上, Benenson 等<sup>[3]</sup>在 2001 年已经利用剪接系统模型成功构造了可编程的有穷自动机. 除此以外, Paun 研究组<sup>[14]</sup>还提出了基于剪接操作的试管系统, 可以实现分布式可编程DNA 计算机. 需要指出的是, 与粘贴模型相比, 剪接模型尽管理论发展比较完善, 但是由于DNA 分子剪接需要酶的参与, 显著提高了成本, 同时由于酶反应本身精确性限制也增加了计算出错的可能.

### 2.3 等价检查模型

等价检查模型是将DNA 双链序列形成的符号串看作两个记忆单元, 然后对这两个单元进行等价性检查. 这种模型的优点是模型简单, 适合分子计算. Yokomori 等<sup>[15]</sup>基于这一模型提出了一种具有图灵机计算能力的DNA-EC 计算机模型, 然而对于该模型的分生物学实验验证尚待进行.

## 3 DNA 计算的应用

DNA 计算机领域取得了飞速的发展, 尽管如此, 研究者普遍认为 DNA 计算机并非作为电子计算机的竞争者或者替代物出现, 它与电子计算机的功能可

能是互补的。由于DNA计算机本身的特点(包括优点和缺点),其今后的发展可能是致力于解决一些电子计算机难以完成的任务。事实上,由于DNA计算机在可靠性、灵活性、可操作性等方面存在的一些问题,我们很难预期其在不久的将来可以实现诸如文字处理、网络等常规功能,然而DNA计算已经在特定的复杂问题领域显示出极大的潜力,从而成为研究者关注的热点。以下将对几个热点应用领域进行介绍。

### 3.1 NP问题

DNA计算机针对NP问题已经有了很出色的表现[1-4],前文所述的HPP问题即属于NP完全问题。NP完全问题还包括图论中的最大连接点问题(maximal clique problem)、布尔代数中SAT问题(satisfiability problem)等。计算机科学已经证明,对所有NP完全问题不存在通用的有效求解算法。Adleman率先通过DNA反应解决了HPP问题,从而开辟了生物计算机的新纪元。其后如SAT问题和最大连接点问题等都被研究者通过DNA计算实现了[2,16]。例如,Liu等[17-22]应用表面化学方法在二维表面上实现了4变量的SAT问题。他们通过限制性酶去除不符合布尔方程的DNA链,然后通过荧光阅读器读出剩余的DNA链即可知道最终解。Landweber研究组[23]则运用一组DNA和RNA实现了9变量的SAT问题,即所谓的象棋中的Knight's问题。最近,Braich等[4]更在试管中实现了20变量的SAT问题。更为重要的是,人们已经意识到生物体内可能就存在很多NP问题,如在一定条件下的蛋白质折叠过程,多重PCR扩增中的引物选择问题等[24],这些可能是通过生物体内天然的DNA计算机来实现的。

### 3.2 遗传分析与临床诊治

DNA计算是通过DNA分子间的特异性反应来实现的。因此利用DNA计算可能反过来对生物学研究起到促进作用,比如对于遗传学和临床检测中非常重要的单核苷酸多态性(single nucleotide polymorphism, SNP)研究[25]。在一定的算法下应用DNA计算过程可以进行高通量的基因分析。2003年日本NovusGene宣称他们已经研制出世界上首台可用于基因分析的DNA计算机并且提供对外服务[26]。尽管他们没有公布该DNA计算机的细节,一般认为他们所报道的是将电子计算机与具有高度并行运算能力的DNA计算单元结合而成的杂合机。尽管这并非真正意义上的DNA计算机,不过这可能是目前最具

实用性的DNA计算机。

应用DNA计算的原理还可以通过作为输入输出单元的DNA分子控制基因表达。Shapiro研究组[5]报道了一种可对mRNA(信使RNA)表达量进行逻辑分析然后据此产生相应分子控制基因表达的DNA分子自动机。该计算机分为3个可编程单元,即分析mRNA量的输入单元;计算单元;产生调控基因表达的DNA分子或治疗药物的输出单元。由于DNA计算机是纳米尺度的计算机,一微升体积中即可包含一万亿个这样的计算机,因此其运算和控制能力是无与伦比的。研究者以肺癌细胞为模型证明了该DNA计算机可以用于体内诊断和在体治疗。

2007年,哈佛大学FAS系统生物学中心的Benenson研究组[27]提出了一种体内DNA计算机的新思路。他们利用了生物体普遍存在的RNA干扰(RNA interfering)机制,根据不同的信号(DNA分子)输入,该DNA计算机可以通过RNA干扰机制实现编码,并据此进行逻辑判断,最后通过一种可发出荧光的蛋白质产生光学信号输出。他们在人肾细胞中初步实现了这一计算过程,显示出DNA计算机具有探测和监控基因突变等细胞内活动的特征信息的能力,因此有可能用于确定癌细胞等病变细胞,成为一种真正的“分子医生”。

### 3.3 防伪和译码技术

DNA分子是天然的海量数据储存载体。由于DNA序列中每个位点都是4种碱基之一,一段20碱基的DNA序列其组合方式可高达一万亿( $4^{20}$ )种。也就是说,当DNA序列达到一定数目之后,其组合方式之多达到了几乎无法穷举的地步。正因为这样,基于DNA高度序列复杂性的DNA防伪认证技术被认为是无法用现有技术破译的终极的防伪技术。Bancroft等[28,29]提出了基于“隐写术”的DNA防伪认证技术,即将防伪密码以特定的方式转换成DNA密码分子并隐藏在其他DNA分子中,然后通过PCR和测序技术可以把这段标记序列阅读出来。他们成立的DNA Technologies公司将其DNA防伪产品成功地运用到2000年悉尼奥运会的特许商品标记上,极大地降低了假冒特许商品造成的损失。Fan等[30,31]近期提出的利用电化学技术的E-DNA生物传感检测技术可以显著降低DNA防伪的成本,可能成为实用化的DNA防伪技术。

在密码破译方面,DNA计算也已经展示了巨大

的应用潜力。DES (data encryption standard system) 密码是一种采用 256 种密钥加密的方法<sup>[32]</sup>，以现有电子计算机的运算能力DES密码是几乎不可能被破译的。然而利用DNA计算机强大的序列编码能力和并行运行能力，研究者表明可以在几个月内获得密码对应的唯一DNA序列。

### 3.4 电脑游戏

DNA计算机最初并没有被考虑应用到游戏领域，然而随着分子自组装领域的飞速发展，研究者开始着手研究可产生运动的分子机器。2003年，Stojanovic<sup>[33]</sup>首次研制出一台具有游戏功能的DNA分子自动机MAYA。他们在研究中应用一类具有酶活性的DNA分子设计成分子逻辑门，并通过23个DNA逻辑门形成布尔网络。1949年在剑桥的EDSAC电子计算机上首次实现了Charles Babbage提出的tic-tac-toe游戏自动机。而MAYA可以完全实现tic-tac-toe游戏的功能，而且可以与人进行交互游戏。有趣的是，由于MAYA所采用的完美策略，在游戏中人总是失败者。尽管MAYA只是实现了最简单的游戏，然而这一步的迈出意味着DNA计算机在游戏领域大有可为，其发展前景不可限量。

## 4 总结与展望

DNA计算机是真正意义上的分子计算机，其诞生和发展具有划时代的意义。与电子计算机中依据一定的法则对有关符号串进行变换的物理过程类似，DNA计算机是通过DNA生物化学性质的变换来实现的。由于DNA计算机是纳米尺度的分子计算机，在很小的体积内可以容纳极大数量的DNA分子，因此拥有无与伦比的并行计算能力。正因为这样，DNA计算机不仅可以解决数学上的难解问题，而且将可应用于研究生命过程的本质问题，即基因组是如何通过对DNA的编码(算法)来实现生物体的生命过程(发育、衰老、疾病、死亡等)的。

DNA计算作为一个新兴的研究领域，已经展现

出非同寻常的发展潜力。然而毋庸置疑，它也面临着各种各样的技术挑战<sup>[6]</sup>。首先是分子生物学实验技术。DNA计算利用的是常规DNA重组技术，然而对于某些环节的要求远较普通分子生物学操作为高。例如，DNA计算所要求的分子连接效率远高于普通DNA重组技术，否则就有可能出现错误答案。另外，目前的DNA计算实验都是采用PCR扩增反应中间物的方法，尚没有很好的方法实现反应中间物的即时分离。这些问题的解决都将依赖于分子生物学操作技术的进一步发展。第二，DNA计算机的芯片化。目前多数的DNA计算机是试管型的，即在溶液中实现。而芯片化的DNA计算机显然更有优势，而且更有可能成为真正的计算机。Zhao等<sup>[34]</sup>最近通过引入磁珠体系，在磁珠表面实现了DNA计算。尽管磁珠体系尚非真正的芯片，这一工作也为DNA计算机最终的芯片化提供了一种有益的思路。第三，数据存取速度问题。尽管DNA计算具有超高的并行计算能力，但是目前的分子生物学操作耗时较长，因而只有在问题的复杂度达到一定程度时才能体现出DNA计算机并行计算的优越性。因此发展DNA序列的快速读取技术是实现数据快速存取的保证。第四，DNA计算机信号的输出。目前的DNA计算机输出信号的检测主要是通过分子生物学操作(如电泳)或荧光阅读器来实现的。而我们知道现在的电子计算机的信号通讯是通过电来实现的。如果能够将计算输出以电信号形式表现，那么DNA计算机与电子计算机之间就能有更好的接口。近期DNA电化学检测技术的快速发展为这个方向提供了一定的可能性。第五，DNA计算机的超级计算能力来源于可以在较小体积内拥有巨大的DNA分子数量。因此，我们预期发展DNA单分子操纵技术将可能最大限度地提升DNA计算机的并行计算能力。然而目前的DNA单分子操作技术主要依赖于扫描探针显微镜(SPM)、单分子荧光光谱等非常昂贵而又难以自动化的技术，这在近期仍将是一个难以解决的矛盾。

## 参考文献

- 1 Adleman L M. Molecular computation of solutions to combinatorial problems. *Science*, 1994, 266(5187): 1021—1024[[doi](#)]
- 2 Ouyang Q, Kaplan P D, Liu S, et al. DNA solution of the maximal clique problem. *Science*, 1997, 278(5337): 446—449[[doi](#)]
- 3 Benenson Y, Paz-Elizur T, Adar R, et al. Programmable and autonomous computing machine made of biomolecules. *Nature*, 2001, 414(6862): 430—434[[doi](#)]
- 4 Braich R S, Chelyapov N, Johnson C, et al. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 2002, 296(5567):

- 499—502[[doi](#)]
- 5 Benenson Y, Gil B, Ben-Dor U, et al. An autonomous molecular computer for logical control of gene expression. *Nature*, 2004, 429(6990): 423—429[[doi](#)]
  - 6 Zhang Z, Fan C, He L. Development of nano-scale DNA computing devices. *Current Nanosci*, 2005, 1: 89—93[[doi](#)]
  - 7 Saghatelian A, Volcker N H, Guckian K M, et al. DNA-based photonic logic gates: AND, NAND, and INHIBIT. *J Am Chem Soc*, 2003, 125: 346—347[[doi](#)]
  - 8 Chen X, Wang Y, Liu Q, et al. Construction of molecular logic gates with a DNA-cleaving deoxyribozyme. *Angew Chem Int Ed*, 2006, 45: 1759—1762[[doi](#)]
  - 9 Xiao X, Hu H. Development of formal models for DNA computing. *Comput Appli*, 2004, 24(9): 123—126
  - 10 Roweis S, Winfree E, Burgoyne R, et al. A sticker-based model for DNA computation. *J Comput Biol*, 1998, 5(4): 615—629
  - 11 Cshaj-Varju E, Freund R, Kari L, et al. DNA computing based on splicing: Universality results. *Pac Symp Biocomput*, 1996: 179—190
  - 12 Kari L, Paun G, Rozenberg G, et al. DNA computing, sticker systems, and universality. *Acta Informatica*, 1998, 35(5): 401—420[[doi](#)]
  - 13 Freund R, Kari L, Paun G. DNA computing based on splicing: The existence of universal computers. *Theor Comput Sys*, 1999, 32(1): 69—112[[doi](#)]
  - 14 Cshaj Varju E, Kari L, Paun G. Test tube distributed systems based on splicing. *Comput Artif Intell*, 1996, 15(2-3): 211—232
  - 15 Yokomori T, Kobayashi S. A model of DNA-computing based on equality checking. *DNA Based Computers III: DIMACS Series*, 1998, 48: 247—260
  - 16 Lipton R J. DNA solution of hard computational problems. *Science*, 1995, 268(5210): 542—545[[doi](#)]
  - 17 Frutos A G, Liu Q, Thiel A J, et al. Demonstration of a word design strategy for DNA computing on surfaces. *Nucleic Acids Res*, 1997, 25(23): 4748—4757[[doi](#)]
  - 18 Liu Q, Frutos A G, Thiel A J, et al. DNA computing on surfaces: Encoding information at the single base level. *J Comput Biol*, 1998, 5(2): 269—278
  - 19 Liu Q, Frutos A G, Wang L, et al. Progress toward demonstration of a surface based DNA computation: A one word approach to solve a model satisfiability problem. *Biosystems*, 1999, 52(1-3): 25—33
  - 20 Liu Q, Wang L, Frutos A G, et al. DNA computing on surfaces. *Nature*, 2000, 403(6766): 175—179[[doi](#)]
  - 21 Smith L M, Corn R M, Condon A E, et al. A surface-based approach to DNA computation. *J Comput Biol*, 1998, 5(2): 255—267
  - 22 Wang L, Liu Q, Frutos A G, et al. Surface-based DNA computing operations: Destroy and Readout. *Biosystems*, 1999, 52(1-3): 189—191[[doi](#)]
  - 23 Cukras A R, Faulhammer D, Lipton R J, et al. Chess games: A model for RNA based computation. *Biosystems*, 1999, 52(1-3): 35—45[[doi](#)]
  - 24 Crescenzi P, Goldman D, Papadimitriou C, et al. On the complexity of protein folding. *J Comput Biol*, 1998, 5(3): 423—465
  - 25 Zhang Z, Zhao J, He L. Progress in molecular biology study of DNA computer. *Acta Genetica Sinica*, 2004, 30(9): 886—892
  - 26 Normile D. Molecular computing: DNA-based computer takes aim at genes. *Science*, 2002, 295(5557): 951[[doi](#)]
  - 27 Rinaudo K, Bleris L, Maddamsetti R, et al. A universal RNAi-based logic evaluator that operates in mammalian cells. *Nat Biotechnol*, 2007, 25: 795—801[[doi](#)]
  - 28 Bancroft C, Bowler T, Bloom B, et al. Long-term storage of information in DNA. *Science*, 2001, 293(5536): 1763—1765[[doi](#)]
  - 29 Clelland C T, Risca V, Bancroft C. Hiding messages in DNA microdots. *Nature*, 1999, 399(6736): 533—534[[doi](#)]
  - 30 Fan C, Plaxco K W, Heeger A J. Electrochemical interrogation of conformational changes as a reagentless method for the sequence-specific detection of DNA. *Proc Natl Acad Sci USA*, 2003, 100(16): 9134—9137[[doi](#)]
  - 31 Fan C, Plaxco K W, Heeger A J. Biosensors based on binding-modulated donor-acceptor distances. *Trends Biotechnol*, 2005, 23(4): 186—192[[doi](#)]
  - 32 Adleman L M, Rothmund P W, Roweis S, et al. On applying molecular computation to the data encryption standard. *J Comput Biol*, 1999, 6(1): 53—63
  - 33 Stojanovic M N, Stefanovic D. A deoxyribozyme-based molecular automaton. *Nat Biotechnol*, 2003, 21(9): 1069—1074[[DOI](#)]
  - 34 赵健, 张治洲, 师咏勇, 等. 运行于磁珠表面的可编程DNA计算机. *科学通报*, 2003, 48(23): 2422—2427