用微型计算机分析核酸序列的初步尝试

夏志清　陈农安
（中国科学院上海生物化学研究所）

计算机处理已成为弄清核酸序列中所包含的大量信息的重要手段。在我国，中、大型的计算机还不多，因此微型计算机的应用就格外重要了。但它字长短（一般是 8 位）；内存小，外设备也有限。本文以 TRS-80I 型 32K 机为例介绍可以在较小的微型机系统中应用，甚至于没有磁盘驱动器的小系统。用这种方法也很容易建立自己的基因库，可以存在磁盘里，也可以存在盒式磁带之中。

一、内存安排

微型机的内存小，是个主要矛盾。这就需要精打细算地安排。核酸序列的每一个碱基占一个字节，字节一一紧密排列起来，成为计算机处理的“数据”，安排在一个特定的区域，有一个固定的起始地址。为了各种不同目的而编写的程序和这种“数据”是脱开的，即数据是独立的。这样使用起来灵活，方便。

在 TRS-80I 型 32 K 机中，以 8801 为始址，BFFF 为终址，可以排一万四千多个碱基，这在目前一般是够用的。将数据安排在高内存区域，便于用 MEMORY SIZE 进行保护。系统占用以及所编写的 BASIC 程序都在低内存区域。如果还有目标码的程序（或子程序）或是其它数据，可放在核酸序列占用区的紧前面（例如，我们的几个目标程序都存放在8800以下）。

BASIC 程序都用 PEEK 函数取得数据。为防止 BASIC 程序挤入数据区，破坏原始信息，应设置保护。

其它以目标码形式组成的程序也可以访问数据区而取得数据。一个核酸序列的数据能为各种不同程序所共用，而同一程序又可处理不同核酸序列。这就是使用的灵活性。

核酸序列中的 A、C、T、G 原可用 ASCII 码或其它形式存入内存；但我们是以 0、1、2、3 来分别代表，这样便于打进去，有一些处理也较方便（下面详述）。为有效地从键盘上打入原始数据，我们编了个“KEYIN”程序，此程序有写入、读出、打印和校对的功能。核酸一级结构序列很长，人工打进去难免有错，而再用人工检查也不易。“CHECK”状态下可重打一次，由计算机核对，可以防止人工打入时产生的错误。

因为 A、T 分别为 0 和 2，C、G 分别为 1 和 3，两组值相减后的绝对值都是 2，利用这种关系可以很容易查出两碱基是否互补。8 位组成一字节，末了第二位代表数值 2，若程序用目标码写成，以位操作改变每个字节的末了第 2 位，则很容易把一段核酸序列改变成相对应的互补序列。

A 和 G、C 和 T 有类似的属性，有的情况需要作这种变换处理。只要注意到 0+3=3；1+2=3 都是等于 3 的这一特点而加以利用就是

另外，DNA 序列中的 T 和 RNA 序列中的 U 都用“2”来代表。

二、几个实例

以下介绍的都以 pBR322 的序列作为其处理对象，此序列有 4632 个碱基对存放在内存 8801—990A 之间。

1、HP1 程序是进行寻找发夹结构（HAIPIN）工作的 BASIC 程序，其中心是寻找互补片断，判断 F 和 H 是否互补，我们用 IFABS (F-H) = 2，(因为如前所述，四种不同碱基是用

0，1，2，3 来表示的)。找到的满足要求的片断在行打上打出来。其形式：

3065 AAACCACCGCTGGTAGCGGTGGTTT

前面的"3065"表明该片断在核酸长链中的位置。

2、RESTM 程序是用 BASIC 语言编写的完成酶切图谱的程序。除和别的程序一样，需要核酸序列作为其数据外，还需要包含一些常用的限制性内切酶的名字和其识别顺序的表。因为这种表还在其它程序中起着作用，我们也使它独立于程序。在内存中是这样安排的：

编 号 表

第 1 种 酶 的 编 号
第 2 种 酶 的 编 号
⋮

每项占 1 个字节

酶 名 表

第 1 种 酶 的 名 字
第 2 种 酶 的 名 字
⋮

每项占 8 个字节

识别顺序表

第 1 种酶的识别顺序	后 继 指 针
第 2 种酶的识别顺序	后 继 指 针
⋮	

7 个字节　1 个字节

此后继指针的含义
若这种酶只有一种识别顺序，为0：若还有其余识别顺序，指向溢出表中相应的后继项位置

溢 出 表

酶的识别顺序	后继指针	指 针
酶的识别顺序	后继指针	指 针
⋮		

7 个字节　1 个字节　1 个字节

后继指针的含义与识别顺序表中的一样，最后一个字节指针指出相应于此识别顺序的酶名及编号在它们各自表中的位置

显然，编号表、酶名表和识别顺序表三种表内的项数是相同的。RESTM 程序对表中所有的酶逐一地在核酸顺序中寻找它的切点来完成酶切图谱。

3、REPT 程序是目标码程序。用来寻找所给核酸序列的重复顺序，以确定重复顺序的所在位置，长短和内容，计算机以各种各样的四联碱基逐个的去试，当发现某四联(例如CAAA)在整个序列中出现多次。选出其中的两段将其邻接部分进行比较，相同则延，不同则停。若有重复顺序，必然可以暴露出来。例如，有如下结果在行打机上输出：

2479 〈T〉〈G〉 AGCAAAAGGCCAG
〈C〉 AA 〈A〉〈A〉 2498

2490 〈C〉〈C〉 AGCAAAAGGCCAG
〈G〉 AA 〈C〉〈C〉 2509

其中括号反映出不相同的碱基。前后的数值则表明了该片断在核酸序列长链中的位置。

4、LPSTM 程序是目标码程序，用来找 RNA Loop STEM 环颈结构可能位置，为研究 RNA 的二级结构提供资料。它与 REPT 程序相似，以四联进行猜试，然后向两方延伸，例如以下的输出结果：

1544 〈A〉〈A〉 CGUGAAGCGA
〈C〉 U 〈G〉〈C〉1559
2044 〈C〉〈A〉 GCACUUCGCU
〈A〉 A 〈G〉〈U〉 2029

其中括号及前后数值同 3 解释。

三、讨　论

1. 按照上述的方法，利用磁盘甚至于磁带，很容易的把已知常用的序列保存起来，形成一个基因库，随时可以调用。

2. 用 BASIC 语言编写程序易掌握，也易于调试，而且又有较强的字符串处理能力，这对核酸序列分析尤为合适。但它占内存大，运行速度慢。当需要循环多次的运行时，就更显得突出。因此，以少数目标程序加于辅助是需要的。

总之，用小系统微型机试上述的方式作为研究核酸的工具效果良好。

[本文于 1982 年 10 月 29 日收到]

日立 835—50 型氨基酸自动分析仪节约茚三酮试剂的一个有效方法

袁　铸　人

（江苏省农业科学院综合实验室）

在氨基酸自动分析时常使用的茚三酮显色剂价格昂贵（每 500g 约 1300 元），消耗量多。我们认为在离子交换层析柱分析完毕一个样品后，当进行再生和平衡时，没有必要再继续供应茚三酮．为此，我们对日立 835—50 型氨基酸自动分析仪分析蛋白质水解液的标准程序（见表 1）作了改变，即在操作工序第 9 步之后加入一条停泵 2 的指令（表 1 中虚线所示），这样，当样品分析到第 48 分钟时，泵 2 停止，亦即茚三酮试剂停止输送，柱的再生和平衡继续由泵 1 来完成。停泵 2 的指令应在样品最后一个组份完全出来后方可执行，以免影响该组份的定量。一般蛋白质水解液样品最后分离出来的是精氨酸，出峰时间约在第 46 分钟，所以我们设置的每个样品分析的终止时间为 48—50 分钟，停泵 2 的时间也定在 48 分钟。

表 1　去氨柱标准程序

操作工序	存储数码	工作数码	时间数码
1	1	1	0
2	2	11	0
3	3	53	0
4	1	2	1
5	4	1	1
6	1	3	8
7	1	4	19
8	1	6	40
9	1	2	46
10	2	10	48
11	1	1	50
12	4	2	65
13	6	2	72

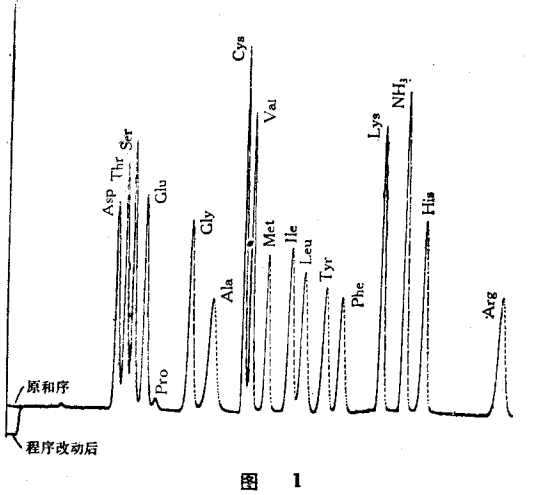

图　1

（下转第32页）