封一

答卷编号（参赛学校填写）：

答卷编号（竞赛组委会填写）：

论文题目： （标明A、B、C、D之一）

B

组 别：(填写研究生、本科生、专科生或中学生**)**

本科生

参赛队员信息(必填)：

|  |  |  |  |
| --- | --- | --- | --- |
|  | **姓 名** | **学 号** | **联系电话** |
| **参赛队员1** | **李沅泽** | **1140310520** | **15114558910** |
| **参赛队员2** | **张翔熙** | **6142910605** | **15645160847** |
| **参赛队员3** | **高琦琦** | **1141900219** | **18245020135** |

参赛学校：哈尔滨工业大学

封二

答卷编号（参赛学校填写）：

答卷编号（竞赛组委会填写）：

评阅情况（学校评阅专家填写）：

学校评阅1.

学校评阅2.

学校评阅3.

评阅情况（联赛评阅专家填写）：

联赛评阅1.

联赛评阅2.

联赛评阅3.

# DNA序列的k-mer index 问题模型

作者：李沅泽 张翔熙 高琦琦

# 摘要：

**关键词：**DNA序列 index trie树 分类算法 k-mer

**1.问题重述**

## 自人类基因组计划实施以来,特别是随着第二代测序技术大规模的使用与推广，现今一次深度测序技术可以产生TB级的序列数据。而在对这些海量数据的分析和应用中，k-mer频次统计是一个非常基础且重要的问题。k-mer频次统计信息可以用来揭示生物各种子序列的分布规律, 它是一种衡量序列相似性的重要工具， 因而其在众多的生物学问题上都有着重要且广泛的应用。【1】

## 部分[DNA](http://baike.baidu.com/view/758.htm)序列或[基因](http://baike.baidu.com/view/8563.htm" \t "_blank)序列使用一串[字母](http://baike.baidu.com/view/94782.htm)表示的真实的或者假设的携带[基因信息](http://baike.baidu.com/view/5178258.htm)的DNA分子的一级结构。可能的[字母](http://baike.baidu.com/view/94782.htm" \t "_blank)只有A，C，G和T，分别代表组成DNA的四种[核苷酸](http://baike.baidu.com/view/117213.htm" \t "_blank)——[腺嘌呤](http://baike.baidu.com/view/62922.htm)，[胞嘧啶](http://baike.baidu.com/view/461057.htm)，[鸟嘌呤](http://baike.baidu.com/view/125105.htm)，[胸腺嘧啶](http://baike.baidu.com/view/476385.htm)。每个[字母](http://baike.baidu.com/view/94782.htm)代表一种碱基，两个碱基形成一个碱基对，碱基对的配对规律是固定的，即是：A-T,C-G。典型的它们无间隔的排列在一起，例如序列AAAGTCTGAC。任意长度大于4的一串[核苷酸](http://baike.baidu.com/view/117213.htm" \t "_blank)被称做一个序列，如 S =“CTGTACTGTAT”。给定一个整数值k，k-mer指的是将一条DNA链，连续切割，挨个碱基划动得到的一序列长度为K的核苷酸序列(如k= 5，则此短串为CTGTA）。

## 现在以文件形式给定 100万个 DNA序列，序列编号为1-1000000，每个基因序列长度为100 。要求对给定k，给出并实现一种数据索引方法，可返回任意一个k-mer所在的DNA序列编号和相应序列中出现的位置。每次建立索引，只需支持一个k值即可，不需要支持全部k值。(如:对序列S来说，所有5-mer为｛CTGTA，TGTAC，GTACT，TACTG，ACTGT，TGTAT｝通常这些k-mer需一种数据索引方法，可被后面的操作快速访问。例如，对5-mer来说，当查询CTGTA，通过这种数据索引方法，可返回其在DNA序列S中的位置为｛1，6｝。)

**》》加入研究背景**

**2.问题分析**

问题分为两个子部分：（1）建立索引，（2）使用索引查询 。要连接这两个子问题，就需要一个中间的索引表Index。建立索引，就是将原始的DNA序列集，转换为索引表。而使用索引查询的过程，就是用指定的k-mer，在索引表中寻找对应的值。

首先要考虑的关键问题，是索引表所包含的信息，以及索引表的数据结构。任何形式的索引表，必须包含每一个 k-mer 所对应的序列编号的集合。注意到 k-mer 共有4^K 种不同的情况，所以索引表必定包含了对应的4K个序列数表Ks。对于每一条长为L的序列S都存在(L-K+1)个子列，极端情况下可能分属 (L-K+1) 个 Ks 中，最大的空间占用长度为 (L-K+1)\* N \* elementSize，最少占用空间不会少于N \* elementSize。

索引表必需的功能有：1.建立一个新索引表; 2.输入k-mer字符串，输出对应的value表的位置或头结点。其中，建立索引表的性能是次要因素，索引表查询的效率是最优先考虑的问题。比如目前的trie树可以在O(n)的时间内完成查询，可能能够完成要求。

综上，我们需要实现的算法包括:

1.空索引表Index的建立

2.空value表的建立

3.通过Index的查询对应的value表位置或头指针

4.value表的写入

5.value表的输出

6.遍历DNA序列S获得(L+1-K)个k-mer子列

建立索引过程为: 1 -> 2 -> 6 -> 3 -> 4；通过索引查询的过程为: 3->5。

2.1.空索引表Index的建立

2.2.空value表的建立

2.3.通过Index的查询对应的value表位置或头指针

2.4.value表的写入

2.5.value表的输出

2.6.遍历DNA序列S获得(L+1-K)个k-mer子列

**3.基本假设**

3.1内存能够容纳至少1个完整的序列记录；

3.2外存能够储存索引表所需的空间；

**4.符号说明**

L：单个DNA序列的长度，即100；

K：子列长度，满足1<=k<=100；

Ks：包含每一个 k-mer 所对应的序列编号的集合；

N：DNA序列总个数，即1000000；

S：DNA序列；

Index：索引表；

elementSize：存储S的序号和子列位置所用的数据类型的空间大小；

li：S中第i个子列的起始位置。

BUF\_LEN：每个数据块大小

Ln：在所有序列中出现过的k-mer子列种类数，亦即键树的叶子总数；

Mk：所有长度为k的ACGT序列的组合数；

MN：在N个长为L的DNA序列中，出现的长度为k的子列的个数；

Node：键树所需的节点个数。

**5.模型建立及求解**

## 建模过程需要实现对DNA序列建立索引，以及搜索过程。现已知数据量较大，为了高效建立索引以及搜索，建立以下三种模型:

1. 模型一

1.1对k-Mer进行编号

#### 用二进制表示每一个碱基:A-->00,T-->11,C-->01,G-->10。对于k-Mer，得到一串01序列，然后将二进制转化为十进制的k-Mer order。如：

#### C T T G A A --> 01 11 10 00 00 --> 480

#### k-Mer -------> k-Mer order

### 1.2把 DNA 序列转换成 k-Mer order 的序列，如：

#### 1326: ACGTTACAGATTTCAGGTTT…

#### 111

#### 1326: ACGTTACAGATTTCAGGTTT…

#### 444

#### 1326: ACGTTACAGATTTCAGGTTT…

#### 753

1.3倒排

#### 对于每一条DNA得到一个数据块，共有一万个数据块。每一项数据定义坐标为 (DNA序号 , k-Mer的起始位置) ，这就完全确定了 k-Mer 的一个位置。再引入一个常数 BUF\_LEN，表示每一个数据块所包含的数据项个数，在之后的操作中，都将.以一个长度为 BUF\_LEN 的数据块为基本单位。

#### 将之前得到的数据，按照 k-Mer order 的不同，分别把位置信息装入不同的数组里。在装入的过程中，一旦某个数组装满，就把这个数组的内容放入文件中，作为一个存储起来的数据块；同时，把对应的 k-Mer order计入另一个文件中，作为数据块顺序的记录。两个文件互相参照，就可以根据k-Mer order出现的位置，到存储数据块的文件中读取对应数据。

|  |  |  |
| --- | --- | --- |
| 排序前的索引文件 | 加入一个序号标记 | 计数排序后的结果 |
| 207 | 1，107 | 7，1 |
| 103 | 2，103 | 235，1 |
| 233 | 3，233 | 23，2 |
| 666 | 4，666 | 46，2 |
| 926 | 5，926 | 19，2 |
| 1023 | 6，1023 | 511，3 |
| 1 | 7，1 | 702，4 |

1.4数据块整理

#### 现在得到的数据已经可以进行搜索了。只需要输入 k-Mer，根据前述方法转化为 k-Mer order数字，然后在存储 k-Mer order 的文件(称为索引文件)里找到出现这个序号的位置，然后在存储数据块的文件(称为数据文件)里找对应位置的数据块，即要查的位置信息。然而，这样的搜索需要遍历整个索引文件，在数据量大的时候效率低。一个解决的方法就是对两个文件进行排序处理，排序后可以有较快的算法进行查找定位。

##### 排序后，按照第一个分量从数据文件中读入对应位置的数据块，依次写入到新的数据文件中，就得到了排序后的数据文件。

##### 然后把所有的第二分量存储起来，就得到了新的索引文件。由于对索引文件和数据文件的改动是同步的，所以对于新获得的两个文件，索引文件中每一条 k-Mer order 数据的位置，都对应着数据文件里一个数据块的位置。

##### 按照之前得到的结果，进行查询的过程为：读入待查的 k-Mer → 转换为对应的 k-Mer order →在索引文件中找到这个 k-Mer order 的位置→ 按照这个位置在数据文件中找到对应的数据块。

##### 改进：考虑建立两个数组，一个用来存储每个 k-Mer order 在索引文件中重复的次数，也就是这个 k-Mer 对应了多少个数据块；另一个数组用来储存每个 k-Mer order出现的位置，也就是对应的数据块在数据文件中的起始位置。(O(logn)的查询过程转化为了O(1))

1.5查询

#### 1.5.1读入要查询的 k-Mer

#### 1.5.2转化为对应的 k-Mer Order

#### 1.5.3在新的索引文件中，查到对应的数据块的数目和起始位置

#### 1.5.4在数据文件中，从起始位置开始，读入指定数目的数据块

#### 1.5.5将 (DNA序号, k-Mer起始位置)形式的数据项转化为DNA序号: (起始位置, 终止位置)的格式，写入到文件中，即得到了查询结果。

1.6问题与缺陷

#### 1.6.1对数据文件重新排序的过程，在增大数据块大小之后，速度明显提升，但导致倒排过程负担过重

#### 改进：可以增加一个中间文件，记录数据块的索引，从而避免直接对数据文件重排。

#### 1.6.2计算 k-Mer Order时，当k大于14，会出现数据类型溢出的问题。既可以通过改用更大的数据类型来解决，也可以自己另行构建一个数据结构。这个数据结构应该有这样的性质：对于所有出现在 DNA 序列里的 k-Mer，都存在一个编号与之对应；对于相同的 k-Mer，具有相同的编号；这个编号应该在int范围内，并且尽量密集分布。

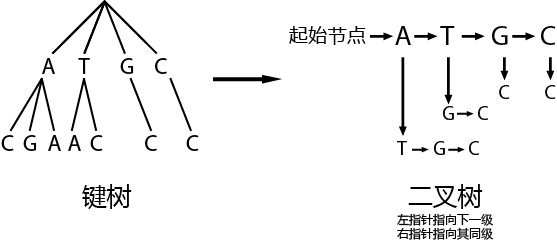
#### 改进：借助一个不完全的四叉树来实现。（参考模型三）

2.模型二——Hash算法

Hash值  
k范围 时间复杂度 对于k的时间

## 3. 模型一——trie树

3.1树的储存



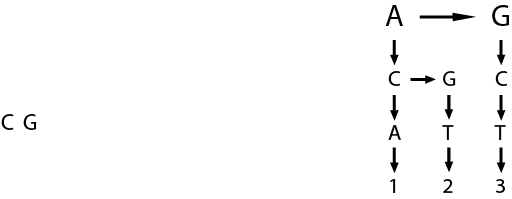
|  |  |  |
| --- | --- | --- |
| int(指向下一级的下标) | char（内容） | int（指向同级的下标） |

节点表示为：

如：tree[3000]

其中，tree[0]不储存元素，表示空节点

|  |  |  |  |
| --- | --- | --- | --- |
| Tree[i] | 下一级 | 内容 | 同级 |
| 0 | 0 | /0 | 0 |
| 1 | 2 | A | 8 |
| 2 | 3 | C | 5 |
| 3 | 4 | A | 0 |
| 4 | 1 | /0 | 0 |
| 5 | 6 | G | 0 |
| 6 | 7 | T | 0 |
| 7 | 2 | /0 | 0 |
| 8 | 9 | G | 0 |
| 9 | 10 | C | 0 |
| 10 | 11 | T | 0 |
| 11 | 3 | /0 | 0 |



建树 -- > 二叉树

**6.模型改进及推广**

对k建立索引，，对1-k所有k值都、能查询

8G内存>>>100万条数据，截成50万，两个树，分别建立索引