ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.2, February 2006, pp.232−241
DOI: 10.1360/jos170232

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于二分频率变换的序列相似性查询处理技术*

王国仁+, 葛 健, 徐恒宇, 郑若石

(东北大学 信息科学与工程学院,辽宁 沈阳 110004)

A Sequence Similarity Query Processing Technique Based on Two-Partitioning Frequency Transformation

WANG Guo-Ren+, GE Jian, XU Heng-Yu, ZHENG Ruo-Shi

(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

+ Corresponding author: Phn: +86-24-83681250, Fax: +86-24-23893138, E-mail: wanggr@mail.neu.edu.cn, http://mitt.neu.edu.cn

Wang GR, Ge J, Xu HY, Zheng RS. A sequence similarity query processing technique based on two-partitioning frequency transformation. *Journal of Software*, 2006,17(2):232−241. http://www.jos.org.cn/1000-9825/17/232.htm

Abstract: As a main method for predicting the functionality of genes, the sequence similarity querying technique is becoming one of the research hotspots in bioinformatics. The similarity of gene sequence and structure usually determines the similarity of gene functionality, and the function of an unknown gene can be predicted by sequence similarity querying. After analyzing the advantages and shortcomings of related work such as frequency transformation and wavelet transformation used in MRS, a new sequence similarity query processing technique based on the two-Partitioning Frequency Transformation 2-PFT is proposed. Firstly, the Two-partitioning frequency transformation and the corresponding distance function are designed. They have a higher filtering ability than frequency transformation and wavelet transformation, and the system performance is thus improved significantly. Secondly, the problem of processing the queries with any length is solved. Theoretical proof and experimental results show that the 2-PFT system outperforms the MRS system greatly.

Key words: sequence similarity query; range query; edit distance; bioinformatics

摘 要: 作为基因功能预测的主要手段,序列相似性查询技术是生物信息学领域的研究热点.基因序列和结构的相似性往往决定了基因功能的相似性,因此可以通过基因序列的相似性查找来预测新基因的功能.分析了 MRS 索引中频率变化和小波变换等相关技术,讨论了它们的缺点和不足,提出了一种基于二分频率变换 2-PFT 的序列相似性查询处理技术.首先,设计了二分频率变换和相应的距离函数,使得系统较之频率变换和小波变换具有更高的过滤能力,极大地提高了系统的性能;其次,解决了处理任意长度查询的问题.理论证明和实验结果均表明,2-PFT 系统的性能远远优于 MRS 系统.

关键词: 序列相似性查询;范围查询;编辑距离;生物信息学
中图法分类号: TP18 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant No.60273079 (国家自然科学基金)
Received 2004-06-20; Accepted 2005-03-11

近几年来,随着基因测序技术的快速发展和人类基因组计划[1]的带动,研究人员对人类及其他生物的基因进行了广泛的测序工作,导致生物序列(包括 DNA、RNA 和蛋白质)信息发生爆炸性的增长.据统计,美国国家生物技术信息中心(NCBI)发布的 Genbank 数据库[2]的数据量正以每 15 个月翻一番的速度增长[3].在这种情况下,生物信息学的出现为处理和分析这些海量生物序列提供了有效的方法.从生物角度来讲,基因结构的相似性往往导致基因功能的相似性.因此,在预测新基因的功能时,我们首先通过序列相似性查询的方法找到与新基因相似的已知基因,然后利用已知基因的功能预测新基因的功能.这就是我们研究序列相似性查询技术的意义所在.随着对基因功能研究的日益深入,序列相似性查询技术逐渐成为生物信息学的研究热点之一.

根据对相似程度的不同要求,序列相似性查询有两类问题:范围查询和 k-近邻查询.范围查询的目的是查找数据库中与查询序列的距离小于查询半径的所有子序列;而 k-近邻查询的目的则是查找与查询序列距离最近的 k 个子序列.我们认为,范围查询比 k-近邻查询更具有实际意义,因为我们对数据库中的生物序列并不了解,也不清楚数据库中的序列与查询序列之间的相似程度.因此我们很难给出 k-近邻查询中适合的 k 值.如果 k 值较小,数据库中可能还有很多的相似子序列被遗漏;如果 k 值较大,得到的多数子序列可能与查询序列的相似程度不高.因此,在多数情况下,k-近邻查询不是很适合.而范围查询则给出了一个相似性的最小值作为查询半径,只要相似度大于该值的子序列都是有意义的查询结果.因此,本文将着重研究范围查询技术,而本文后面提到的序列相似性查询即指范围查询.下面给出了范围查找的定义.

给定生物序列数据库 $S=\{S_1,S_2,\ldots,S_d\}$,其中,$S_1,S_2,\ldots,S_d$ 是数据库中的基因序列.范围查询是指对于给定的错误率 e 和查询序列 q,找到数据库 S 中所有与 q 的距离小于半径 r 的所有子序列,其中查询半径 r 为错误率 e 和查询序列长度的乘积,即 $r=e*|q|$.基因序列在计算机中表示为由字母 A,C,G,T,N 组成的一个字符串,因为基因序列就是由这几个字母所代表的核苷酸组成的(N 为未确定的核苷酸位置).而序列间的相似性则是由编辑距离[4]来衡量的.

现有的序列相似性查询技术分为非索引方法和基于索引的方法两类.非索引方法往往需要搜索整个数据库,因此其性能很不理想;而最近提出的基于索引的方法性能较好.然而,现有的基于索引的方法还存在很多问题:(1) 这些方法在建立索引过程中使用的变换方法丢失了大量位置信息,从而导致其距离函数的过滤能力不高,整个系统的性能不理想;(2) 现有系统都不能处理任意长度的查询,只能处理查询序列的一个较长的前缀.

针对这些问题,本文提出了一种基于二分频率变换(two-partitioning frequency transformation)的序列相似性查询技术 2-PFT.首先,我们提出了二分频率变换技术,将字符串变换为高维空间中的向量.这种变换比现有的频率变换和小波变换保留了更多的位置信息,从而使基于这种变换的距离函数有更好的过滤能力;其次,我们针对处理任意长度查询序列的问题提出了解决方案;最后,我们还提出了一种新的查询分片策略,进一步提高了系统的性能.我们不仅给出了严格的理论证明,而且进行了大量的实验.实验结果表明,我们的技术在各种查询半径下都优于现有的技术,而且性能有几倍至十几倍的提高.

1 相关工作

计算编辑距离是序列模式相似性问题最直接的解决方法.但是,因为计算编辑距离所消耗的时间和空间都很大,因此研究人员提出了很多改进的方法[5−7].这些方法的特点是不使用索引结构,因此统称为非索引方法.非索引方法往往需要遍历整个数据库,因此,随着数据序列的不断增长,这些方法越来越不能满足要求.近年来出现的几种方法[8−10]普遍使用了索引技术,通过访问索引结构,过滤掉大量的不相关记录,从而极大地减少了数据库的访问次数和编辑距离的计算次数,提高了性能.这类方法统称为基于索引的方法.

基于索引的方法的一般流程是:首先将数据库中特定长度的子序列通过某种变换,转化为高维空间中的向量,然后再利用某种高维空间的索引结构对其进行索引.这种索引结构应该足够小,以便于存放在内存中.同时,根据使用的变换,定义一种向量之间的距离函数代替编辑距离,对不相关记录进行过滤.为保证查询结果的完整性,这种距离函数必须是编辑距离的下限.用户提交查询序列和查询半径后,系统首先通过索引结构进行查询,对不相关记录进行过滤,得到一个候选结果集合.之后,根据候选结果集合访问数据库,利用编辑距离对候选集

合中的结果进行验证,得到最终结果.

MRS 系统[8]是目前性能最好的系统之一.在对字符串进行变换的过程中,该系统提出了一种称为“频率变换”的变换方法,即将特定长度的字符串转换为字符串中各个字符出现次数的频率向量.例如,字符串 S=“ACAGTT”,其中 A 出现两次,C,G 各出现一次,T 出现两次,N 出现 0 次,因此变换之后得到的频率向量 $T(S)$=(2,1,1,2,0).为了提高系统的过滤能力,该系统又提出了一种“小波变换”,其变换的结果是两个 5 维的向量.第 1 个向量相当于对原字符串进行一次频率变换;而第 2 个向量相当于把原字符串平均分成两个子字符串,并将两个子串分别进行频率变换,再将这两个子串的频率向量作差.

之后,该系统定义了基于上面两种变换的距离函数,用于代替编辑距离,对不相关记录进行过滤.为了定义距离函数,该系统首先定义了正、负距离的概念.对于向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$,它们之间的正、负距离定义如下:

正距离:

$$Pos=\sum_{u_i>v_i}u_i-v_i \tag{1}$$

负距离:

$$Neg=\sum_{u_i<v_i}v_i-u_i \tag{2}$$

对于频率变换所得到的向量,其距离公式为

$$FD_1(\boldsymbol{u},\boldsymbol{v})=\max(Pos,Neg) \tag{3}$$

对于小波变换,其距离公式为

$$FD_2(\boldsymbol{u},\boldsymbol{v})=\begin{cases}\dfrac{|Pos-Neg|}{2}, & \text{if} \quad \min<\dfrac{|Pos-Neg|}{2}\\ \dfrac{|Pos-Neg|}{2}+\dfrac{\min-\dfrac{|Pos-Neg|}{2}}{2}, & \text{else}\end{cases} \tag{4}$$

该系统使用 MBR(minimal bounding rectangle)将变换后得到的向量索引起来,存储在内存中.因为距离函数体现的是在两个等长字符串之间的距离,因此,为了处理任意长度的查询序列,该系统用长度为 2^a,2^{a+1},2^{a+2},…,2^b 的滑动窗口建立索引,形成一种多解析度的索引结构.在进行查询时,首先将查询序列分片为长度为 2^a,2^{a+1},2^{a+2},…,2^b 的若干个子查询(最末端长度小于 2^a 的片段会被忽略).对于每个子查询,该系统先将其变换为频率向量,再选择长度相等的索引层次,利用上述的距离函数计算每个 MBR 与查询向量之间的距离.如果某个 MBR 与查询向量的距离大于查询半径,该 MBR 就被过滤掉.所有子查询都处理过以后,留下来的 MBR 集合称为候选结果集合.后处理阶段根据候选结果集合访问数据库,计算每个 MBR 对应的子序列与查询序列之间的编辑距离,从而得到最终结果.

总之,该系统通过基于索引的查询过程,过滤掉大量的不相关结果,极大地减少了数据库的访问次数和编辑距离的计算次数,从而提高了查询性能.

CoMRI 系统[9]是在 MRS 系统基础上提出的.在建立索引阶段,CoMRI 系统提出了一种 k-元组频率变换方法[10].它不是简单地记录每个字符出现的次数,而是记录每个 k-元组出现的次数.并提出了相应的距离公式.同时,提出 VBR 结构[9]代替原来的 MBR 作为索引结构.但是,该系统提出的距离函数不能保证小于编辑距离.这样,虽然其过滤性能很好,但是可能错误地过滤掉很多正确的结果,导致查询结果不完整.因此,我们认为该系统是不可取的.我们希望在保证正确性的基础上,尽量提高查询性能.

2 2-PFT 系统

由上面的分析可知,整个查询过程消耗的时间由两部分组成:基于索引进行查询的时间和后处理阶段计算编辑距离进行验证的时间.

在基于索引的查询过程中,影响性能的关键因素在于距离函数的过滤能力.因为,每次子查询都是在前面的子查询所得结果的基础上进行的.因此,如果距离函数的过滤能力较好,那么就能够在每次子查询过程中过滤掉

大量不相关记录,减少中间结果,这样,后面的子查询的计算量就会大为减少,从而提高查询系统的性能.

对于后处理阶段中计算编辑距离对候选结果集合进行验证的过程,其性能完全取决于编辑距离的计算次数.很明显,候选结果集合越小,需要计算编辑距离的次数越少.因此从根本上讲,这一问题同样与距离函数的过滤能力密切相关.由此可见,距离函数的过滤能力是影响整个查询系统性能的关键因素.而距离函数的定义是与系统所使用的变换密切相关的.变换保留原字符串的信息越多,距离函数就越接近于编辑距离,该距离函数的过滤能力也就越强.当然,更好的查询技术和后处理技术同样可以大幅度地提高系统的性能.

2.1 二分频率变换和相应的距离函数

MRS系统中使用的频率变换,直接将序列中某个字符出现的次数累加作为向量中对应维的值.很显然,这种变换丢失了原字符串的位置信息.也就是说,它不关心某个字符出现的位置,仅关心该字符出现的次数,这样会导致原本相差很大的两个字符串,变换之后得到相同的频率向量.例如,字符串 S_1="ACGT",字符串 S_2="TCAG",那么变换之后得到的频率向量 $T(S_1)=(1,1,1,1,0)$,$T(S_2)=(1,1,1,1,0)$,很明显,其距离 $FD_1=0$.而两个字符串之间实际的编辑距离 $ED(S_1,S_2)=3$.由此可见,这种变换丢失了原来字符串的位置信息,导致由其计算得到的距离和实际的编辑距离有较大的差距.

在小波变换的过程中,虽然将字符串分为两段,保留了一定的位置信息,但这种变换又将两段的频率向量作差.这种做法没有实际意义,而且破坏了位置信息,导致其距离函数不是非常理想.总之,由于 MRS 系统中提出的变换丢失了原字符串的全部或大部分位置信息,导致其距离函数的过滤能力较差,从而使得候选结果集合较大,基于索引的查询时间和后处理时间都较长,算法的性能不是很好.

为了保留原字符串的全部字符信息和位置信息,我们提出了 N 分频率变换的概念.它将长度为 N 的字符串分为 N 段,即每个字符属于一段.然后对每段(也就是每个字符)作频率变换,得到 N 个 5 维频率向量.例如,给定字符串 S="ACAGTT",经过变换后得到一个 6×5 维的向量,即 6 个 5 维的向量:(1,0,0,0,0),(0,1,0,0,0),(1,0,0,0,0),(0,0,1,0,0),(0,0,0,1,0),(0,0,0,0,1).这样,这种变换不仅保留了字符信息,更重要的是保留了全部的位置信息.例如,第 3 个向量的第 1 个分量值为 1,说明原字符串的第 3 个位置上的字符为"A",而第 4 个向量的第 3 个分量的值为 1,说明字符串的第 4 个字符为"G".

但是,这种变换的最大缺点是,它占用了过多的存储空间.如果一条字符串的长度为 N,那么使用这种变换需要存储 N 个 5 维向量即 $5N$ 个整数.而 MRS 系统只需要存储两个 5 维向量即 N 个或 $2N$ 个个整数即可.如果字符串的长度很长,要索引的字符串个数很多,我们就很难保证索引结构能够存储在内存中.

为了使我们的索引结构占用的空间不超过MRS系统,又能够保留一定的字符串位置信息,我们取 $N=2$,得到了一种二分频率变换,即 2-PFT.也就是说,它首先将字符串分为前后两段,然后对这两个子字符串分别进行频率变换,得到两个 5 维的向量.

给定字符串 S,其长度为 N,其字母来源于包含 k 个字母的字母表 $\Sigma=\{\alpha_1,\alpha_2,\ldots,\alpha_k\}$.其二分频率变换 2-PFT 定义为 $2\text{-}PFT(S)=(\boldsymbol{A},\boldsymbol{B})$,其中,$\boldsymbol{A}=(\alpha_1,\alpha_2,\ldots,\alpha_k)=FT(S[0{:}n/2-1])$,$\boldsymbol{B}=(b_1,b_2,\ldots,b_k)=FT(S[n/2{:}n-1])$,其中 FT 为前述的经典频率变换.

例如,字符串 S="ACGT",那么我们首先将 S 分为 S_1="AC"和 S_2="GT"两段,然后分别对 S_1 和 S_2 进行频率变换,得到两个 5 维向量:$\boldsymbol{A}=(1,1,0,0,0)$;$\boldsymbol{B}=(0,0,1,1,0)$,因此,$(\boldsymbol{A},\boldsymbol{B})$即为我们的变换结果.

这样一来,虽然我们没有保留全部的位置信息,但是我们保留了前、后两段的位置信息.因此,我们的变换保留的信息比 MRS 系统要多,我们的距离函数就会更接近编辑距离,其过滤能力也更好,从而提高了算法的性能.更重要的是,我们的索引结构并没有占用更多的存储空间,它同样可以放入内存中,大大提高了在索引上进行查询的效率.

下面,我们首先定义对字符串的一次编辑操作对其二分频率向量的影响,然后定义基于二分频率变换的距离函数.

定理 1. 给定一个字符串 S,S 中的字符来源于字母表 $\Sigma=\{\alpha_1,\alpha_2,\ldots,\alpha_k\}$.字符串 S 的二分频率向量 $2\text{-}PFT(S)=(\boldsymbol{A},\boldsymbol{B})$.其中,$\boldsymbol{A}=(\alpha_1,\alpha_2,\ldots,\alpha_k)$,$\boldsymbol{B}=(b_1,b_2,\ldots,b_k)$.那么,对字符串 S 进行的一次编辑操作可能对其二分频率向

量($\boldsymbol{A}$,$\boldsymbol{B}$)产生如下影响:

(1) $a_i=a_i+1,a_j=a_j-1$; (2) $b_i=b_i+1,b_j=b_j-1$;

(3) $a_i=a_i\pm1$; (4) $b_i=b_i\pm1$;

(5) $a_i=a_i+1,b_i=b_i-1,b_j=b_j+1$; (6) $a_i=a_i-1,b_i=b_i+1,b_j=b_j-1$;

(7) $a_i=a_i-1,a_j=a_j+1,b_i=b_i+1$; (8) $a_i=a_i+1,a_j=a_j-1,b_i=b_i-1$.

上述(1)项、(2)项相当于对 S 进行一次替换操作.如果替换发生在 S 的前半部分,那么会产生(1)的结果;如果替换发生在后半部分就得到(2)的结果.例如,字符串 S="ACGT",我们对 S 进行一次替换操作,将字符 C 替换为字符 T,得到 ATGT.其原来的二分频率向量为 $\boldsymbol{A}$=(1,1,0,0),$\boldsymbol{B}$=(0,0,1,1);而替换后,其二分频率向量变为 $\boldsymbol{A}$=(1,0,0,1),$\boldsymbol{B}$=(0,0,1,1),即(1)的情况.同理,当我们将字符串"ACGT"中的"T"变换为"A"得到"ACGA"时,就会发生(2)中所示的情况.这两项操作对向量的影响为(+1,−1).

(3)项、(4)项相当于对 S 进行一次插入或删除操作.其中的"+"对应于插入操作,"−"对应于删除操作.如果插入或删除操作发生在 S 的前半部分,那么会产生(3)的结果;如果发生在后半部分就产生(4)的结果.例如,字符串 S="CGT",我们在最前端插入字符"A"得到"ACGT".那么,原来的二分频率向量为 $\boldsymbol{A}$=(0,1,0,0),$\boldsymbol{B}$=(0,0,1,1);而替换后,其二分频率向量变为 $\boldsymbol{A}$=(1,1,0,0),$\boldsymbol{B}$=(0,0,1,1),即情况(3)所示的变化.同理,当我们在字符串"ACGT"的后面插入一个"A"得到"ACGTA"时,就会发生(4)中所示的情况.这两项操作对向量的影响为(±1,0).

(5)~(8)项同样对应于一次插入或删除操作,但是这种插入、删除操作不同于(3),(4)项中的操作,因为它引起了原字符串的重新分段.例如,我们将字符串"ACACT"变为"ACACTT".原来的分段为"AC"和"ACT",经变换后得到 $\boldsymbol{A}$=(1,1,0,0),$\boldsymbol{B}$=(1,1,0,1);而插入一个字符"T"后,分段将变为"ACA"和"CTT",从而得到 $\boldsymbol{A}$=(2,1,0,0),$\boldsymbol{B}$=(0,1,0,2),即(5)所示的情况.如果将字符串"ACGCT"变为"AACGCT"就会发生(7)所示的情况.删除的操作也是同样道理.(5)和(7)对应于插入操作,对向量的影响为(+2,−1);而(6)和(8)对应于删除操作,其作用相当于(+1,−2).

设现有两个向量 $\boldsymbol{u}$,$\boldsymbol{v}$,已知其正距离为 Pos,负距离为 Neg,我们要计算它们之间的二分频率距离.二分频率距离定义为:用定理 1 中的操作将向量 $\boldsymbol{u}$ 变成向量 $\boldsymbol{v}$ 所需要的最少的操作次数.也就是说,将其正、负距离调整为 0 所需的最少的操作次数.由定理 1 可知,操作(5)~(8)对向量的影响作用最大,因此,应该尽可能多使用这些操作.

假设 Pos 大于 Neg,我们每次使用(5)~(8)的操作时,如果 Neg 减少 1,那么 Pos 将减少 2.因此,如果 $Pos>2Neg$,那么,我们就可以使用 Neg 次的(5)~(8)操作,将 Neg 调整为 0.此时,剩余的 Pos 为 $Pos-2Neg$.我们再用($Pos-2Neg$)次的操作(3),(4)就可以将 Pos 调整为 0.因此,总的操作次数为:$Neg+(Pos-2Neg)$,即 $Pos-Neg$.如果 $Pos\le2Neg$,那么我们首先作($Pos-Neg$)次的操作(5)~(8),使剩余的 $Pos=Neg=Neg-(Pos-Neg)=2Neg-Pos$.之后,再利用($2Neg-Pos$)次的操作(1),(2),将剩余的 Pos 和 Neg 同时调整为 0.因此,总的操作次数为($Pos-Neg$)+($2Neg-Pos$)=Neg.同理可知 Pos 小于等于 Neg 的情况.

综上所述,我们得到二分频率距离函数:

$$2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})=\begin{cases}|Pos-Neg|, & \text{if } \min<|Pos-Neg|\\ \min, & \text{else}\end{cases} \tag{5}$$

二分频率距离是比频率距离 FD_1和小波距离 FD_2更接近编辑距离的一种距离函数.因为它保留了更多的位置信息.举例来说,给定字符串 S_1=ACGT,字符串 S_2=TCAG,经过频率变换之后得到的频率向量 $FT(S_1)$=(1,1,1,1),$FT(S_2)$=(1,1,1,1),其距离 $FD_1(S_1,S_2)=0$,$ED_1(S_1,S_2)=0$.而经过二分频率变换后得到 $2\text{-}PFT(S_1)$=((1,1,0,0);(0,0,1,1)),$2\text{-}PFT(S_2)$=((0,1,0,1);(1,0,1,0)),利用式(2)得到其二分频率距离 $2\text{-}PFD=2$.可见,我们提出的二分频率距离的确更接近于编辑距离.

下面,我们从理论上证明我们的二分频率距离 $2\text{-}PFD$ 比频率距离和小波距离更接近于编辑距离.我们首先定义计算向量间正、负距离的函数,然后根据引理 1 简化各种距离的计算过程,最后分别给出"$FD_1\le2\text{-}PFD$"和"$FD_2\le2\text{-}PFD$"的证明过程.

由式(3)、式(4)以及式(5)给出的距离计算公式可以看出,这些距离函数都是基于正、负距离的.也就是说,

我们必须首先计算两个向量之间的正、负距离,然后根据上述几个公式计算它们之间的各种距离.因此,我们希望给出计算正、负距离的函数,以便于进行证明.

由式(1)、式(2)我们可以看到,计算正、负距离的过程相当于先计算两个向量的差向量,然后将大于 0 的维累加即为正距离,而小于 0 的维累加的绝对值即为负距离.因此,我们给出计算正、负距离的另一种形式的函数:

给定向量 $\boldsymbol{u}$ 和向量 $\boldsymbol{v}$,那么向量 $\boldsymbol{u}$,$\boldsymbol{v}$ 之间的正距离 $Pos(\boldsymbol{u},\boldsymbol{v})$和负距离 $Neg(\boldsymbol{u},\boldsymbol{v})$的距离函数如式(6)、式(7)所示.

$$Pos(\boldsymbol{u},\boldsymbol{v})=PS(\boldsymbol{u}-\boldsymbol{v}) \tag{6}$$

$$Neg(\boldsymbol{u},\boldsymbol{v})=NS(\boldsymbol{u}-\boldsymbol{v}) \tag{7}$$

其中,函数 $PS(\boldsymbol{X})$定义为向量 $\boldsymbol{X}$ 中正数维的和,$NS(\boldsymbol{X})$定义为向量 $\boldsymbol{X}$ 中负数的和,如式(8)、式(9)所示.

$$PS(\boldsymbol{X})=\sum_{X_i>0}X_i \tag{8}$$

$$NS(\boldsymbol{X})=\sum_{X_i<0}X_i \tag{9}$$

下面,我们证明一个频率变换的性质作为引理,然后简化各种距离函数.

引理 **1**. 给定字符串 S_1 和 S_2,经频率变换得 $FT(S_1)=\boldsymbol{u}=(u_1,u_2,\ldots,u_k)$,$FT(S_2)=\boldsymbol{v}=(v_1,v_2,\ldots,v_k)$.由式(1)、式(2)计算得到正、负距离为 Pos_1 和 Neg_1;经小波变换后得到正、负距离 Pos_2 和 Neg_2;经二分频率变换后得到正、负距离 Pos_3 和 Neg_3.当字符串 S_1 和 S_2 长度相同时,这些变换都有如下性质:正距离等于负距离,即 $Pos_1=Neg_1$,$Pos_2=Neg_2$,$Pos_3=Neg_3$.证明如下.

证明:因为$|S_1|=|S_2|$,所以,$\sum_{i=1}^{k}u_i=|S_1|$,$\sum_{i=1}^{k}v_i=|S_2|$,故此,$\sum_{i=1}^{k}u_i=\sum_{i=1}^{k}v_i$.

删除两边相等的项,然后将那些小于 u_i 的 v_i 移到等号左侧,将小于 v_i 的 u_i 移到等号右侧,得到:

$$\sum_{u_i>v_i}(u_i-v_i)=\sum_{v_i>u_i}(v_i-u_i).$$

即 $Pos_1=Neg_1$.同理可得,$Pos_2=Neg_2$,$Pos_3=Neg_3$. □

根据引理 1 和公式(3)~(5),我们可以对上面各种距离函数进行简化,得到:

$$FD_1=Pos_1;\ FD_2=Pos_2/2;\ 2\text{-}PFD=Pos_3 \tag{10}$$

下面,我们首先给出函数 PS 的性质,如引理 2,然后,证明二分频率距离 2-PFD 大于等于频率距离 FD_1.

引理 **2**. 对于经过频率变换得到的两个向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$,函数 PS 有如下性质:

$$PS(\boldsymbol{u}+\boldsymbol{v})\le PS(\boldsymbol{u})+PS(\boldsymbol{v}).$$

证明:对于 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 的任意一维 u_i 和 v_i,我们作如下讨论:

i) $u_i\ge 0$,$u_i\ge 0$

$(u_i+v_i)=(u_i+v_i)$,则在该维上,$PS(\boldsymbol{u}+\boldsymbol{v})=PS(u)+PS(v)$;

ii) $u_i\ge 0$,$u_i\le 0$

$(u_i+v_i)\le u_i$,则在该维上,$PS(\boldsymbol{u}+\boldsymbol{v})\le PS(\boldsymbol{u})+PS(\boldsymbol{v})$.

同理可证其他两种情况. □

定理 **2**. 设经过二分频率变换后得到的两个向量 $\boldsymbol{u}=(\boldsymbol{A}_1,\boldsymbol{B}_1)$,$\boldsymbol{v}=(\boldsymbol{A}_2,\boldsymbol{B}_2)$,其中 $\boldsymbol{A}_1,\boldsymbol{B}_1,\boldsymbol{A}_2,\boldsymbol{B}_2$ 均为向量,分别对应两个字符串的前、后两部分的频率向量.那么,经过频率变换得到的两个向量为 $\boldsymbol{s}=(\boldsymbol{A}_1+\boldsymbol{B}_1)$,$\boldsymbol{t}=(\boldsymbol{A}_2+\boldsymbol{B}_2)$,则 $FD_1(\boldsymbol{s},\boldsymbol{t})\le 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$.

证明:根据式(6)和式(10)有:

$2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})=PS(\boldsymbol{u}-\boldsymbol{v})=PS(\boldsymbol{A}_1-\boldsymbol{A}_2)+PS(\boldsymbol{B}_1-\boldsymbol{B}_2)$; ①

$FD_1(\boldsymbol{s},\boldsymbol{t})=PS((\boldsymbol{A}_1+\boldsymbol{B}_1)-(\boldsymbol{A}_2+\boldsymbol{B}_2))=PS((\boldsymbol{A}_1-\boldsymbol{A}_2)+(\boldsymbol{B}_1-\boldsymbol{B}_2))$;

再根据引理 2 得:

$FD_1(\boldsymbol{s},\boldsymbol{t})\le PS(\boldsymbol{A}_1-\boldsymbol{A}_2)+PS(\boldsymbol{B}_1-\boldsymbol{B}_2)$; ②

综合式①、式②得:$FD_1(\boldsymbol{s},\boldsymbol{t})\le 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$. □

下面,我们给出函数 PS 的另一个性质——引理 3,之后证明二分频率距离 2-*PFD* 大于等于小波距离 FD_2.

引理 **3**. 对于任意两个经过频率变换得到的向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$,总有 $PS(\boldsymbol{u}-\boldsymbol{v})=PS(\boldsymbol{v}-\boldsymbol{u})$.

证明:因为向量$(\boldsymbol{v}-\boldsymbol{u})=-(\boldsymbol{u}-\boldsymbol{v})$,因此,$Pos(\boldsymbol{u}-\boldsymbol{v})=Neg(\boldsymbol{v}-\boldsymbol{u})$,而且 $Neg(\boldsymbol{u}-\boldsymbol{v})=Pos(\boldsymbol{v}-\boldsymbol{u})$.

根据引理 1 可知,$Pos(\boldsymbol{u}-\boldsymbol{v})=Neg(\boldsymbol{u}-\boldsymbol{v})$,而且 $Pos(\boldsymbol{v}-\boldsymbol{u})=Neg(\boldsymbol{v}-\boldsymbol{u})$.

因此,$PS(\boldsymbol{u}-\boldsymbol{v})=PS(\boldsymbol{v}-\boldsymbol{u})$. □

定理 **3**. 设经过二分频率变换后得到的两个向量 $\boldsymbol{u}=(\boldsymbol{A}_1,\boldsymbol{B}_1),\boldsymbol{v}=(\boldsymbol{A}_2,\boldsymbol{B}_2)$,其中 $\boldsymbol{A}_1,\boldsymbol{B}_1,\boldsymbol{A}_2,\boldsymbol{B}_2$ 均为向量,分别对应两个字符串的前、后两部分的频率向量.那么,经过小波变换得到的两个向量为 $\boldsymbol{s}=((\boldsymbol{A}_1+\boldsymbol{B}_1),(\boldsymbol{A}_1-\boldsymbol{B}_1))$, $\boldsymbol{t}=((\boldsymbol{A}_2+\boldsymbol{B}_2),(\boldsymbol{A}_2-\boldsymbol{B}_2))$,则 $FD_2(\boldsymbol{s},\boldsymbol{t})\le 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$.

证明:

$2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})=PS(\boldsymbol{u}-\boldsymbol{v})=PS(\boldsymbol{A}_1-\boldsymbol{A}_2)+PS(\boldsymbol{B}_1-\boldsymbol{B}_2)$; ①

$FD_2(\boldsymbol{s},\boldsymbol{t})=PS(\boldsymbol{s}-\boldsymbol{t})/2=[PS((\boldsymbol{A}_1+\boldsymbol{B}_1)-(\boldsymbol{A}_2+\boldsymbol{B}_2))+PS((\boldsymbol{A}_1-\boldsymbol{B}_1)-(\boldsymbol{A}_2-\boldsymbol{B}_2))]/2$;

令 $L=PS((\boldsymbol{A}_1+\boldsymbol{B}_1)-(\boldsymbol{A}_2+\boldsymbol{B}_2)),R=PS((\boldsymbol{A}_1-\boldsymbol{B}_1)-(\boldsymbol{A}_2-\boldsymbol{B}_2))$,那么,由定理 2 可知,

$L\le 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$; ②

$R=PS((\boldsymbol{A}_1-\boldsymbol{B}_1)-(\boldsymbol{A}_2-\boldsymbol{B}_2))=PS((\boldsymbol{A}_1-\boldsymbol{A}_2)+(\boldsymbol{B}_2-\boldsymbol{B}_1))$;

由引理 2 可知:$R\le PS(\boldsymbol{A}_1-\boldsymbol{A}_2)+PS(\boldsymbol{B}_2-\boldsymbol{B}_1)$;又根据引理 3 可知:$PS(\boldsymbol{B}_2-\boldsymbol{B}_1)=PS(\boldsymbol{B}_1-\boldsymbol{B}_2)$;

所以,

$R\le PS(\boldsymbol{A}_1-\boldsymbol{A}_2)+PS(\boldsymbol{B}_1-\boldsymbol{B}_2)$; ③

综合式①~③可知:

$L+R\le 2\times 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$;即$(L+R)/2\le 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$;即 $FD_2(\boldsymbol{s},\boldsymbol{t})\le 2\text{-}PFD(\boldsymbol{u},\boldsymbol{v})$. □

2.2 处理任意长度的查询序列

如前所述,MRS 系统能够处理的最短查询序列的长度为 2^a,因此它只能处理长度为 2^a 的倍数的查询序列.而对于分片后剩余的不足 2^a 的部分,将被系统忽略.因此,MRS 系统不能处理任意长度的查询序列.

在 MRS 系统中长度不足 2^a 的子查询不能被系统处理,这是因为,式(1)、式(2)定义的是由长度相等的字符串变换所得的向量之间的正、负距离,式(3)、式(4)则是在正、负距离基础上定义的.而在系统的索引中,最短长度的子序列为 2^a,因此,无法应用式(1)~(4)所定义的距离函数来处理长度不足 2^a 的子查询.

以下我们分析了长度对距离的影响,并在此基础上改进了长度不相等时向量之间的正、负距离计算方法,从而解决了处理任意长度查询序列的问题.

给定长度不同的字符串 S_1 和 S_2,长度分别为 a 和 b.经二分频率变换得到向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$.由式(1)、式(2)计算得到正、负距离为 *Pos* 和 *Neg*.首先假设 $a<b$,根据正、负距离的计算方法可知:如果两个字符串的长度相等,那么正距离表达了将向量 $\boldsymbol{v}$ 转化为向量 $\boldsymbol{u}$ 所需要的插入操作的次数;而负距离表达了将向量 $\boldsymbol{v}$ 转化为向量 $\boldsymbol{u}$ 所需的删除操作的次数.因此,如果 $a<b$,那么会导致将向量 $\boldsymbol{v}$ 变成向量 $\boldsymbol{u}$ 所需的删除操作次数变大,即 *Neg* 的值会增大,而增大的部分就是两个字符串长度的差 $b-a$;如果 $a>b$,那么会导致将向量 $\boldsymbol{v}$ 变成向量 $\boldsymbol{u}$ 所需的插入操作次数变大,即 *Pos* 的值会增大,而增大的部分就是两个字符串长度的差 $a-b$.

根据上面的分析,我们只要根据两个字符串长度的差异对正、负距离进行调整,就可以得到真正有意义的正、负距离.如算法 1 所示.

算法 **1**. 长度不等时正、负距离的调整算法.

Input:

Pos,*Neg* /* 由式(1)、式(2)计算得到的正、负距离 */

a,*b* /* 两个字符串的长度 */

Output:

Pos,*Neg* /* 调整后的正、负距离 */

Algorithm body:

If ($a<b$)
    Neg–=(b–a);
elseif ($a>b$)
    Pos–=(a–b);
Endif

例如,假设在我们的系统中,索引的最小长度为 128,而分片后剩余的子查询的长度为 100,而且用式(1)、式(2)计算得到的正距离 Pos=38,负距离 Neg=10.根据我们的分析和算法 1 可知,由于子查询的长度小于索引的长度,导致正距离增大,而增大的幅度为两者的长度差,即(128–100).因此,真正有意义的正距离为 38–(128–100)=10.

这样,我们的系统就真正解决了处理任意长度查询序列的问题.假设 a=7,b=10,即子查询的可选长度为 128,256,512 和 1024,那么按照 MRS 系统的查询分片策略,长度为 2 768 的查询序列的分片结果如图 1 所示.

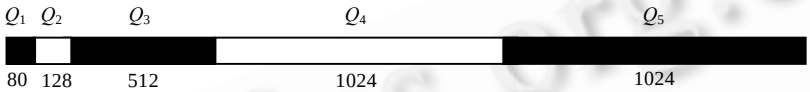

Fig.1 Query partitioning strategy in MRS system
图 1 MRS 系统的查询分片策略

文献[8]提到,字符串之间的编辑距离和频率距离,随着两个字符串长度的增大而增大.因此,对于长度比较长的子序列,通过距离函数计算得到的距离会比较大,也因此更容易被排除掉.根据这一发现,如果我们改变分片策略,使最长的查询片段作为第 1 个子查询,最短的查询片段作为最后一个子查询,那么就可以尽早排除更多的不相关记录,极大地减少中间结果,从而缩短了基于索引的查询时间,也减少了候选结果集合的大小,提高了系统的性能.

基于这种考虑,我们提出了一种新的分片策略,即我们总是从查询序列的前面开始分片,每次尽可能分得最长的子查询.这样就保证了较长的子查询优先处理.基于这种分片策略,长度为 2 768 的查询序列的分片结果如图 2 所示.

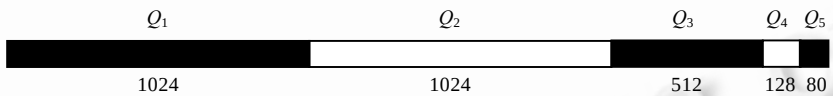

Fig.2 A new query partitioning strategy
图 2 新的查询分片策略

3 性能测试与分析

为了测试基于二分频率变换的距离函数的过滤能力以及 2-PFT 系统的性能,我们在 SUN Enterprise-450 服务器上分别实现了 N-PFT 系统和 MRS 系统,并分别针对两个系统的 3 个主要性能指标(过滤能力、基于索引的查询时间以及总查询时间)进行了测试和对比.

在测试过程中,我们选择人类的 18 号染色体和 22 号染色体序列作为测试数据,并从该染色体序列中随机抽取了 6~8 组长度不等的序列作为查询序列.在建立索引过程中选取参数 a=7,b=10,MBR 的容量设置为 200.因为范围查询的查询半径对查询系统的响应时间有较大影响,因此,对于任何一个性能指标,我们分别测试其在不同染色体序列、不同查询半径下的性能.其中,查询半径由错误率 e 决定,而错误率 e 在[0.01,0.05]范围内变化.

对于同一染色体序列、同一查询半径,我们使用不同的查询序列进行多组测试,并将各组结果的均值作为最终结果.

3.1 过滤能力

过滤能力是影响系统性能的主要因素.对于过滤能力的测试,我们主要通过候选结果集合中 MBR 的个数来衡量.这是因为,过滤能力越强,过滤掉的记录越多,候选结果集合中留下来的 MBR 个数就越少.MRS 系统和

2-PFT 系统的过滤能力对比如图 3、图 4 所示.

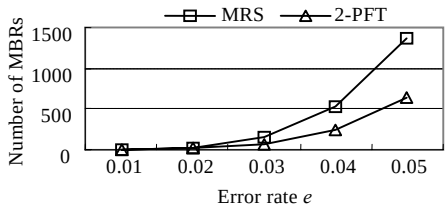

Fig.3 The pruning ability of *N*-PFT system and MRS system on CHR18

图 3 过滤能力的比较——染色体 18

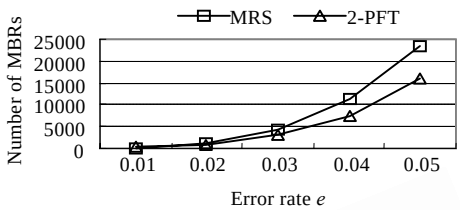

Fig.4 The pruning ability of *N*-PFT system and MRS system on CHR22

图 4 过滤能力的比较——染色体 22

可见,基于任何测试序列、任何查询半径,基于二分频率距离的 2-PFT 系统的候选结果集合中 MBR 的个数都比 MRS 系统更少.这表明,我们提出的基于二分频率变换的距离函数的过滤能力在各种条件下都优于 MRS 系统中的距离函数.这主要是因为,我们提出的二分频率变换保留了更多的信息,从而使我们的距离函数更接近于编辑距离.

3.2 基于索引的查询时间

为了测试新的距离函数和分片方法对基于索引的查询性能的影响,我们测试了基于索引的查询时间,即从提交查询到得到候选结果集合的过程所消耗的时间.其性能对比如图 5、图 6 所示.

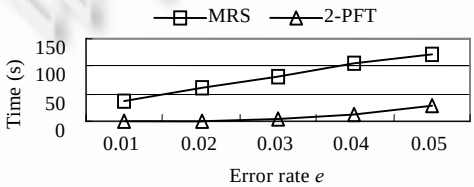

Fig.5 The cost for index based query of 2-PFT system and MRS system on CHR18

图 5 基于索引的查询时间——染色体 18

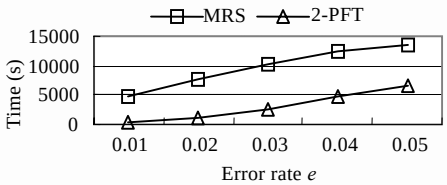

Fig.6 The cost for index based query of 2-PFT system and MRS system on CHR22

图 6 基于索引的查询时间——染色体 22

通过上面的测试结果,我们可以看到,2-PFT 系统极大地缩短了基于索引的查询时间.在染色体 18 上,2-PFT 系统的基于索引的查询性能比 MRS 系统提高了 4~70 倍.在染色体 22 上也有 2~20 倍的提高.

2-PFT 系统极大地提高了基于索引的查询性能,这是因为:① 距离函数的过滤能力较强;② 新的分片策略使较长的子查询优先处理.在这两个因素的作用下,2-PFT 系统极大地减少了查询过程的中间结果,从而减少了后面子查询的计算量,缩短了查询时间.

3.3 总查询时间

下面给出系统总查询时间的对比曲线,如图 7、图 8 所示.

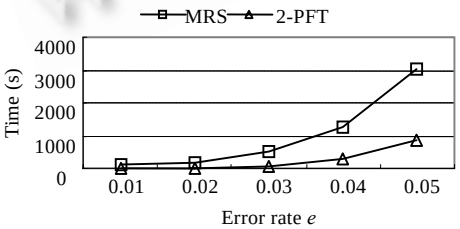

Fig.7 The total cost of 2-PFT system and MRS system on CHR18

图 7 总查询时间——染色体 18

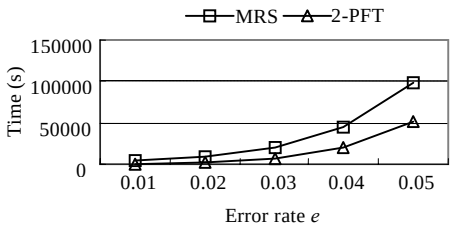

Fig.8 The total cost of 2-PFT system and MRS system on CHR22

图 8 总查询时间——染色体 22

由于 2-PFT 系统在基于索引的查询时间方面优于 MRS 系统,而且其候选结果集合更小,后处理过程消耗的时间必然少于 MRS 系统,因此,2-PFT 系统的总查询时间必然优于 MRS 系统.而且,在各种错误率下其性能分别提高了 2~15 倍左右.

4 结束语

本文主要研究生物序列数据库中序列相似性查询技术.针对现有系统的缺点,本文做了如下几方面的工作:① 提出 N 分频率变换技术,保留了字符信息和全部的位置信息;② 考虑到索引大小的问题,采用了二分频率变换,并提出了一种过滤能力更高的距离函数;③ 解决了处理任意长度查询的问题;④ 提出一种新的查询分片策略,进一步提高了系统的性能.理论证明和实验分析表明,我们提出的 N-PFT 系统的平均性能比现有的 MRS 系统提高了 2~15 倍.

我们下一步的工作主要集中在:① 从理论上证明 N 值越大,相应的距离函数的过滤能力越强;② 研究 N 的取值与索引大小的关系,从而在保证索引能够存放在内存中的情况下,尽量增大 N 的取值;③ 提供序列相似性查询的在线服务.

References:

[1] The human genome project (HGP). 2006. http://www.nhgri.nih.gov/

[2] National Center for Biotechnology Information. Genbank database. 2005. http://www.ncbi.nlm.nih.gov/

[3] Benson DA, Karsh-Mizrachi I, Lipman DJ, Ostell J, Rapp BA, Wheeler DL. Genbank. Nucleic Acids Research, 2000,28(1): 15−18.

[4] Gusfield D. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge: Cambridge University Press, 1997.

[5] Myers E. An $O(ND)$ difference algorithm and its variations. Algorithmica, 1986,1(2):251−266.

[6] Myers E. A sublinear algorithm for approximate keyword matching. Algorithmica, 1994,12(4-5):345−374.

[7] Baeza-Yates RA, Navarro G. Faster approximate string matching. Algorithmica, 1999,23(2):127−158.

[8] Kahveci T, Singh AK. An efficient index structure for string databases. In: Apers P, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass R, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB 2001). Roma: Morgan Kaufmann Publishers, 2001. 351−360.

[9] Sun H, Ozturk O, Ferhatosmanoglu H. CoMRI: A compressed muti-resolution index structure for sequence similarity queries. In: Peter M, Xu Y, ed. Proc. of the 2nd IEEE Computer Society Bioinformatics Conf. (CSB 2003). Califonia: IEEE Computer Society, 2003. 553−559.

[10] Ferhatosmanoglu H, Ozturk O. Effective indexing and filtering for similarity search in large biosequence databases. In: Jamil H, Megalooikonomou V, ed. Proc. of the 3rd IEEE Int'l Symp. on BioInformatics and BioEngineering (BIBE 2003). Washington DC: IEEE Computer Society, 2003. 359−366.

王国仁(1966－),男,湖北崇阳人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为 XML 数据管理,生物信息学,P2P 数据管理,多媒体索引技术.

徐恒宇(1978－),男,硕士,主要研究领域为生物信息学.

葛健(1978－),男,硕士,主要研究领域为生物信息学.

郑若石(1980－),男,硕士,主要研究领域为生物信息学.