文章编号：2095-2163(2019)01-0001-06 中图分类号：TP391.2 文献标志码：A

基于 de Bruijn 图的基因组索引结构设计

国宏哲，王亚东
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要： 随着高通量测序技术的快速发展和测序成本的逐渐降低，个体基因组测序已成为研究不同物种的基因型、变异情况和相关疾病的重要手段。然而，由于基因组上的大量重复序列和高变异区域，日益增大的测序数据量以及测序技术的局限等因素，如何准确且快速地将大量测序数据比对到参考基因组面临巨大挑战。阐述基于哈希思想的基因组数据的存储和索引方法。本文说明基于 seed-and-extension 思想的基本比对思路。本文提出一个基于 de Bruijn 图模型的索引结构 DBG-index 以及该索引的 3 层结构数据存储方式。分析该索引结构的特性并提出种子的基本操作方法。该索引结构利用图模型特性可以有效组织基因组上的重复序列，从而在整体上减少了候选种子数量并极大提高了比对速度。

关键词： 基因组；索引；序列映射；de Bruijn 图

Design of de Bruijn graph-based genome indexing structure

GUO Hongzhe, WANG Yadong
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 With the rapid development of sequencing technology and its gradual cost reduction, individual genome sequencing has become the main approach to study the genotypes of different species, variation knowledge and the related diseases. However, due to the massive repetitive sequences and high complex genomic regions, the ever-increasing sequencing data size and the technical limitations of sequencing technology, how to effectively and efficiently map the amount of reads to reference genomes is still facing the great challenges. This thesis introduces the hash table-based genomic data storage and indexing method and the basic idea of seed-and-extension scheme. A de Bruijn graph-based indexing structure named as DBG-index and its three-level storage mode are proposed. Moreover, several basic corresponding operations are put forward based on the index characteristics. It demonstrates that this structure could effectively organize and index the repetitive sequences on the genomes in such a way that the number of candidate seeds could be decreased and the mapping speed could greatly increase.

【Key words】 genome; index; reads mapping; de Bruijn graph

0 引 言

HTS 技术的一项重要挑战是将测序序列快速且准确地比对到一个或多个参考基因组上。随着 HTS 技术的普及而产生的数据量越来越大，序列比对是 HTS 分析中计算量最大的步骤之一。read 比对过程关于速度的要求正日渐提升。而且随着越来越多的新基因组的陆续面世，更新的 HTS 数据分析需要将 reads 比对到多个基因组、而不是单基因组上。一个物种的典型链的基因组可以作为金标准，再利用来源于宏基因组样本的 reads 比对到这些参考基因组去识别物种类型和病原体类型[1-3]。同时，de Bruijn 图是代表和组织基因组序列的基本数据结构，并已广泛应用于各类序列分析的学术领域中，如 de novo 基因组装、高通量测序(NGS)序列比对、泛基因组分析、宏基因组学分类、转录本同种型鉴定和定量分析、NGS 序列校正等。这些也是许多基因组学研究的基础。

时下，多数位居前沿的先进比对软件都是根据单基因组进行设计并优化的。同时也有很多方法被提出索引 pan-genome[4-5]，且都是通过共线性多序列比对技术组织基因组。大部分索引 pan-genome 的方法只能用于小规模的变异研究，例如软件 GenomeMapper [4] 和 BWBBLE[5] 只能整合 SNP 和短 Indel 变异。同时，基于共线性结构的比对速度也较慢，其速度比目前较快的比对软件慢几十倍[5-6]。该方法的适用范围仅限于高度相似的多基因组，但还不能针对来源于不同链或不同物种的多基因组发挥效用。这些非同源性的基因组中经常包含结构变异且共线性结构不能提供诸如大型结构变异的非线

基金项目： 国家重点研发计划(2017YFC0907503)。

作者简介： 国宏哲(1987-)，男，博士研究生，主要研究方向：生物信息学；王亚东(1964-)，男，教授，博士生导师，主要研究方向：生物信息学、人工智能、知识工程等。

收稿日期： 2018-10-26

性序列变换。共线性结构本身也影响序列比对结果,不适合将序列比对到多个不同基因组。产生式压缩后缀数组索引[6]会随着变异数量增加而呈指数增长,使其很难处理在重复区域的变异信息[7]。增广多基因图结构(population reference graph, PRG)[8]可以更好支持不同种类的变异。该方法是将所有 reads 构建成 de Bruijn 图和 PRG 进行比对。其本质是将基因组序列比对和从头拼接相结合。该方法主要是为局部区域而设计适合解决基因组上复杂区域上的相关问题并提高了基因组复杂区域的比对质量。但目前尚未见到有效的方法索引全基因组的 PRG 结构,还不能将此方法延伸至全基因组上的序列比对。为解决以上方法的限制并优化处理基因组上大量重复片段引起的问题,本文提出一个基于 de Bruijn 图的索引结构 DBG-index 来有效组织重复序列,从而实现单基因组和多基因组的索引和序列比对计算。

1 基于 de Bruijn 图的索引结构分析

1.1 基于哈希表结构的数据存储方法

De Bruijn 图结构可以利用 hash table 数据结构进行存储,即将各个 k-mer 节点提取指定 bp 数的 2-bit 位表示(A/C/G/T 分别编码成二进制 00/01/10/11)作为地址空间中的地址值。该地址记录 k-mer 对应入边信息和出边信息,及指向其对应的位置信息的地址,如图 1 所示。利用 de Bruijn 图构建索引并实现序列比对的优势如下。某一 k-mer 序列在基因组中可能出现在多个位置上,de Bruijn 图中某一节点表示的 k-mer 只出现一次且该节点对应的所有位置信息记录在一个数据结构中。在基于后缀树结构和基于 BWT 结构构建的索引中,每个后缀在基因组中对应唯一的一个位置,其位置信息是分散的,但 de Bruijn 图中节点将位置信息集合在一起且允许有序。可以进行相邻节点对应位置集合的合并操作以快速筛选提取真实位置信息,并验证当前路径是否正确。通过对现在映射系统原理的分析,发现基于 seed-and-extension 思想方法[9-11]比对过程中对候选位置进行的 extension 局部比对是相对耗费时间的计算步骤。如何有效地缩小候选位置的范围、减少 extension 的次数对系统速度实现至关重要。而基于 de Bruijn 图结构的遍历过程会大大减少候选 extension 的位置集合数量。

de Bruijn 图结构的一个特点是可以实现双向图,即某一个节点既可以记录其出边,也可以记录其入边(基因组上 k-mer 序列对应的上一个 bp)。该特性可以使种子的延伸方法更加灵活,可以采用双向延伸的方法,从而更快速获得较长的种子和真实的候选位置。

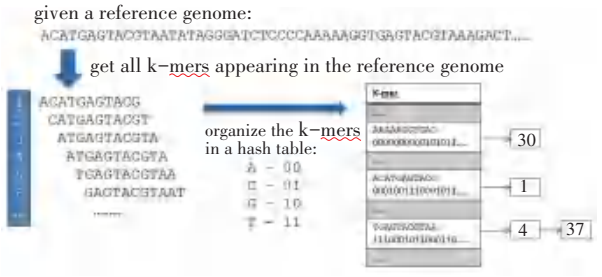

图 1 基于 Hash 结构的基因组存储方式

Fig. 1 An example of genomic k-mers storage in the hash table structure

1.2 基于哈希结构的序列比对方法

k-mer 长度越长,在比对过程中需要搜索 k-mer 的次数越少,且速度也越快。但在实际应用中需要严格控制 k-mer 的长度,因为地址空间是随着其序列长度的增加而呈指数增长。而且,k-mer 长度过大会使比对时 seed 过长而较大概率地跨过 mismatch 或 indel 存在的 bp 位置,将真实的 seed 漏掉而无法得到真实的位置信息。如果 k-mer 长度过小会使整个 de Bruijn 图过密、节点过多,且每个节点对应的入边和出边也较多,这会使图上的搜索过慢。故 k-mer 长度的选取成为一个关键问题。针对此问题,需要计算不同 k-mer 长度在不同物种基因组上的分布。

很多物种的基因组数据的 k-mer 长度变化趋势,开始快速增长到一个值后趋于稳定。其中,hg18 人类基因组数据快速增长到 20 后,开始缓慢增长,一直到 40 后趋于稳定。

结合 Hash 实现过程中的实际情况,针对人类基因组数据使用,本文将序列映射系统索引和比对过程中 k-mer 长度范围定为 21~28,将前 14 bp 作为地址,并将剩余 bp 构建连续的数组结构(地址空间约 $2^{28} * 4\ B = 1\ GB$)。

基于 Hash 索引结构进行序列比对的基本原理如图 2 所示。从 read 上从头向后提取 k-mer 序列转换成地址编码,在 Hash 索引中进行搜索获得对应的获选位置,在对应位置提取序列和 read 做局部或全局比对。这也是基于 seed-and-extension 的经典流程。

基于 de Bruijn 图的比对过程与此有所不同,在获得某节点的位置信息后会有合并过程,通过此有

序位置数据间的合并过程可筛去大量无意义的位置信息，如图 3 所示。某一位置集合里的位置数值加上 read 上偏移，在相邻节点对应位置集合中搜索找到合理的位置数值，从而实现两集合间的合并，按照此方法依次向下合并。在合并过程中考虑到 mismatch 或 indel 存在的可能性，根据测序序列的错误率和各个类型变异的发生概率设定一个误差范围。

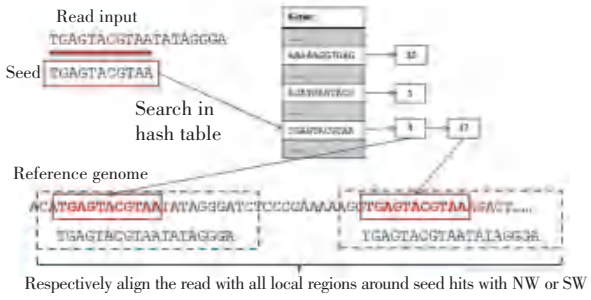

图 2　基于 hash 结构存储的 read 比对过程

Fig. 2　An illustration of the read alignment based on the hash structure

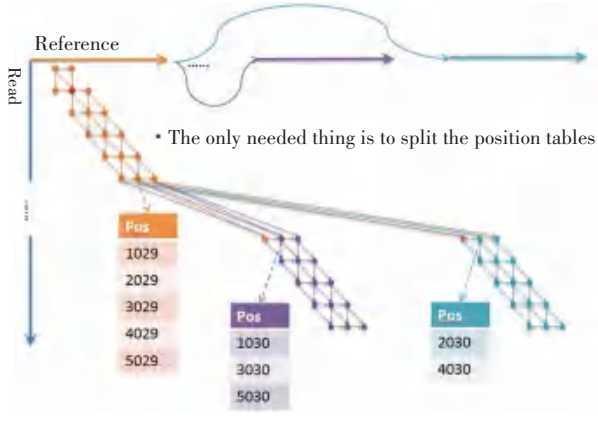

图 3　种子对应位置集合合并过程

Fig. 3　An illustration of the process of seeds merging by the position collections

1.3　k-mer 的 3 层存储结构分析

各部分索引结构设计思想如图 4 所示。其中包含一个 3 层结构：第一层是 hash 的地址空间保存 k-mer 的前 14 bp 序列信息；第二层记录后面剩余的 k-mer 信息，空余出来的信息位可用于记录该 k-mer 是否有反转操作等信息（用于双向 seed 延伸操作使用）及入边和出边信息；第三层是 k-mer 对应的有序位置信息数组。

为实现该索引结构，系统需要遍历一遍基因组序列。且第二层序列内容存储和第三层位置信息需要有序存储，故需要先将基因组上所有 k-mer 进行排序。而由于基因组序列较长无法做到将所有 k-mer 统一提取排序，故采用分段排序的办法、即按照字母表逻辑大小顺序（A<C<G<T）先把 k-mer 信息写入临时文件，例如，以 AAAA 开头的 k-mer 写入第一个临时文件中；以 AAAC 开头的 k-mer 写入第一个临时文件中；以 TTTT 开头的 k-mer 写入最后一个临时文件中。临时文件具体个数根据索引构建过程调试结果而定。所有临时文件间是有序的，即第一个临时文件中所有 k-mer 逻辑顺序全部小于第二个临时文件，以此类推。

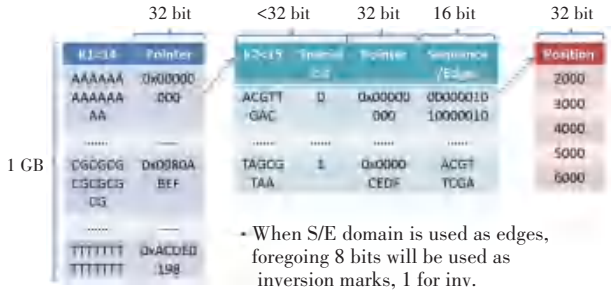

图 4　索引存储三层结构

Fig. 4　An illustration of three-level indexing structure

需要注意在临时文件中也记录了 k-mer 对应的位置偏移信息及其对应的入边和出边信息。然后按顺序重新读取每个临时文件，即某一时刻只有一个临时文件的全部 k-mer 调取至内存中，对这些 k-mer 生成排序。在此过程中也可以同时构建索引的 3 层结构：每当临时文件中 k-mer 排完序后从头遍历该有序结构，每当遇到一个不同 k-mer 序列，即将该 k-mer 添加到 Hash 地址空间中并添加入边和出边信息；若当前 k-mer 序列与前一个 k-mer 序列相同，则在 k-mer 序列对应的位置信息地址后添加新位置信息。

在比对部分首先导入索引文件。某一 k-mer 的搜索在第一层地址空间中时间是 $O(1)$；在第二层后半部分，k-mer 序列搜索过程时间是 $O(n)$；第三层返回的位置信息数组是由第二层结构中的指针加以控制。其做到了最快速地返回某一 k-mer 的所有相关信息并实现了位置信息的融合且有序。基于此结构的 read 的 seeding 过程可阐释如下。

（1）从 read 头入手，提取 k-mer 作为 seed 开始在 Hash 索引结构上根据各节点位置集合做延伸停止后，从停止处的下一个 bp 开始再延伸。

（2）按此方式一直到 read 遍历完成，将所有获得的 seed 对应的位置结合再做进一步合并，检验是否有 seed 间可以合并的可能。

（3）将最终得到的 seed 按种子长度逐一排序，依次提取 seed 对应位置集合中位置在基因组上的序列做局部比对或全局比对；若某 2 个相邻 seed 得

到的比对分数收敛或设定某一固定的结束条件,则终止局部比对的 extension 环节输出比对结果。

2 索引结构构建方法

2.1 索引的基本组织结构分析

本文构建一个(Reference de Bruijn Graph)结构去组织一个或多个 reference 并用到基于 hash table 的数据索引结构 DBG-index 去索引 de Bruijn 图上的所有 unipaths,而不是索引原始基因组。DBG-index 组织和索引 reference 示例,一个或多个 reference 被组织到。一个 k-mer 的所有拷贝被定位到 de Bruijn 图上的同一个节点,基因组上序列相同的重复片段被定位到相同的 unipath 上。该新基因组的索引方法提供 2 个基本操作去识别和合并相似种子,并且利用 de Bruijn 图上 unipath 的性质识别相同的局部序列。此处 unipath 定义满足如下条件:路径起始节点的入度大于 1、且出度是 1;路径的结束节点入度是 1、且出度大于 1;路径中间节点入度是 1、且出度是 1。索引可以同时处理多个相似种子和相同局部序列,从而有效解决基因组的重复性带来的问题。

比对系统是利用索引相关功能设计推出了在图索引上的基于 seed-and-extension 思想的序列映射算法。算法很好处理基因组上的重复片段,从而极大减少了序列比对的时间开销。用户可以根据自定义 k-mer 长度构建 reference 的 de Bruijn 图。de Bruijn 图上的所有节点和边是通过 reference 上的所有 k+1-mers 计算获得。遍历 de Bruijn 图的 hash table 结构,若当前 k-mer 为一个起始节点、即开始根据其出边获得其相邻的下一个 k-mer,在 hash table 结构中定位该 k-mer,并按此方法循环直至当前 k-mer 为结束节点,此时可形成一个 unipath 序列并开始下一个 unipath 的生成。按此方式遍历完 hash table 即可生成所有的 unipath。一个 unipath 在基因组上可能有多个拷贝,故需要记录每一个 unipath 对应的所有拷贝在基因组上的真实起始位置。索引结构包括以下 3 个主要结构:线性表记录所有 unipath 序列且为每一个 unipath 指定一个标识;一个 hash table 索引 unipath 上的所有 k-mers;线性表记录每个 unipath 的所有拷贝的起始位置。而且通过该索引结构可实现给定 k-mer 获得其所在的 unipath 和在 unipath 的偏移位置;给定一个 unipath 和某偏移位置获得在基因组上原始拷贝位置集合。

2.2 相似种子合并操作方法

根据 de Bruijn 图的原理,某一个 k-mer 在图结构上只出现一次,任意一个指定 k-mer 所在的 unipath 和坐标是唯一的。且某个 k-mer 在基因组上的位置集合可通过线性表联合计算获得。利用以上功能接口可以通过实现以下 2 个主要操作(相同 unipath 上的相似 seeds 合并;unipath 上相同序列合并)去合并相似种子、以及减少相同的 extension 计算。

某一 k-mer 在 reference 上的所有拷贝位于图结构上的同一节点,该性质表明一个 k-bp 长的 seed 在 reference 上的映射等同于连接 seed 和图结构上的特定节点。seed 的命中是一段短片段精确比对到一个 unipath 的多个拷贝上。通过以上数据结构可计算某拷贝在 unipath 上的起始位置。

某一 read 映射到基因组上的某一可能位置可以被估算成某 unipath 拷贝在基因组上位置加上对应 seed 在 unipath 上的位置偏移。多个 seeds 在同一个 unipath 则可能有 2 个相似的候选位置集合且可以将相似位置集合予以合并。可以利用线性表判断任意 2 个 seed 是否映射到同一个 unipath,如果是同一 unipath,可以通过发现 seeds 间的相似性从而进行相似 seeds 的合并而不需要分别计算每一个命中位置。在此基础上,考察某一 seed 在其 unipath 上的偏移和到 unipath 的 2 个端点的距离,从而判断该 unipath 序列是否容纳 read 序列进行局部比对。如果可以,unipath 对应的所有拷贝和 read 的所有 extension 计算就都将归结为 read 和 unipath 序列本身的局部比对,而不需要对 seed 的每一个命中分别进行 extension 计算。

3 结果分析

为测试基于索引结构的 seeding(种子获取过程)效果,研究分别统计模拟数据和真实测序数据上的 seeding 中各过程产生的 seeds 数,并使用 BWA 比对软件中的基于 BWT 索引结构的索引和基于 BWT MEM 索引结构(maximal exact matches,MEM)的索引进行比较分析。实验中拟使用 Mason 模拟器分别模拟生成基于 Illumina 平台的长度为 100 bp 和 250 bp 的 reads 的数据集(Sim-i100 和 Sim-i250)。

模拟数据集上的索引结构 seeding 统计结果和测序数据集上的索引结构 seeding 统计结果可分别详见表 1 和表 2。由表 1 可见,索引结构对应的 seeding 过程中第一部分表示初始产生的 seeds 个

数；第二部分表示相同 unipath 上 seeds 合并后的 seeds 数；第三部分表示 unipath 上的相同序列合并后的 seeds 数。BWT MEM 索引对应的 seeding 过程中第一部分表示初始产生的 seeds 数；第二部分表示经过 MEM 计算后的 seeds 数。Filtering 过程表示各索引结构自身定义的一些 seeds 的过滤规则，例如，索引中将最长 seed 长度小于当前 read 长度一半的 seeds 过滤掉。不同的过滤规则筛选出的 seeds 会有所不同。模拟数据集上 *Valued seeds ratio* 定义为产生正确比对位置的 seeds 占经过 filtering 后的 seeds 总数的比例；测序数据集上 *Valued seeds ratio* 定义为产生合理比对位置的 seeds 占经过 filtering 后的 seeds 总数的比例。

表 1　模拟数据集上的索引结构 seeding 统计结果分析

Tab. 1　Statistical analysis of DBG－index based seeding on simulation dataset

Dataset	Method	Seeding #/ ×10^6		Filtering #/ ×10^6		*Valued seeds ratio*/%
Sim-i100	DBG-index	4.90	4.60	4.50	4.10	48.50
	BWT-index		178.90		178.90	1.10
	BWT MEM-index	32.60		19.20	12.50	15.90
Sim-i250	DBG-index	11.30	9.90	8.90	5.70	34.50
	BWT-index		871.70		871.70	0.23
	BWT MEM-index		68.40	44.30	8.10	24.80

表 2　测序数据集上的索引结构 seeding 统计结果

Tab. 2　Statistics of DBG-index based seeding on sequencing dataset

Dataset	Method	Seeding #/ ×10^7		Filtering #/ ×10^7		*Valued seeds ratio*/%
ERR174324	DBG-index	1.53	1.51	1.50	1.20	15.70
	BWT-index		277.90		277.90	0.07
	BWT-MEM-index		2.15	1.76	1.69	11.80

由表 1 和表 2 可看出，各索引结构随着各个计算过程 seeds 逐渐减少。模拟数据集和真实数据集上 A 索引在各个过程生成的 seeds 最少且最后的 *Valued seeds ratio* 最高。BWT 索引在各个过程生成的 seeds 最多且最后的 *Valued seeds ratio* 最低。因为 MEM 方法有种子长度扩展和筛选功能，BWT MEM 索引相对 BWT 索引生成更少 seeds。A 索引的 seeding 的有效性是 BWT 索引的几十倍。相对于 BWT MEM 索引生成的 seeds 数目，A 索引同样表现出优势，且在 Sim-i100 上效果优势更加明显。因为 read 长度较短时可用的 seeds 数也相对较少，由此即说明当 seeds 数较少时，索引能更快速地生成正确 seeds。这对于后续能够高效实现序列映射算法应用将十分重要。

研究使用来源于千人基因组计划中的样本 NA12878 的 Illumina 测序平台生成的 ERR174324 真实数据集测试各索引结构 seeding 情况。由于真实数据中的测序错误，也加入了更多、更复杂变异，从表 2 可看出，整体上真实数据比模拟数据上产生更多的 seeds。所有的 *Valued seeds ratio* 相对模拟数据降低，同样由于测序错误等问题使索引生成很多错误 seeds。在该情况下，索引仍然表现出较好的 seeding 结果。

在 seeding 过程中每个 seed 都有其对应的命中(hits)数，表示在基因组上对应的位置个数。由于基因组上有大量重复片段，一个 seed 通常对应多个 hits。在映射算法中考虑到速度的要求，需要对 hit 设定一个阈值以实现整个系统在准确度或敏感度和速度上的平衡。为此，研究将在测序数据集上测试不同 hits 数对 seeding 情况的影响。研究得出人类基因组上的相关统计结果见表 3。其中，Hits-n 表示 hits 数上限阈值是 n。随着 n 的增大，seeds 数增大且 *Valued seeds ratio* 降低。可以观察到在 Hits-300～-700 范围内，seeds 数增长平稳，且在 Hits 数大于 1 000 时 seeds 数趋于平稳。此类统计分析有利于辅助映射算法选择默认 hits 阈值。

表 3　测序数据集上索引结构的不同 hits 数量阈值下 seeding 统计结果

Tab. 3　Statistics of DBG－index based seeding with various hits value thresholds on sequencing dataset

statistics	Hits-100	Hits-300	Hits-500	Hits-700	Hits-900	Hits-1 100
Seeds #/×10^7	1.16	1.27	1.31	1.34	1.37	1.37
Valued seeds ratio/ %	17.20	15.70	15.20	14.90	14.50	14.50

4　结束语

本文首先阐述基于 hash table 结构的一般基因组索引结构和基于此结构的比对方法。接下来分析了不同长度的 k-mer 在基因组上的分布情况。然后探讨了 seed-and-extension 的基本思路，并发现根据位置集合对种子进行合并可以有效减少 extension 的计算量。

本文首次提出基于 hash 思想的用于 k-mer 存储的 3 层索引结构，并以此为基础提出基于 de Bruijn图的索引结构来组织基因组中的重复序列。

（下转第 13 页）