第二篇
高通量测序数据分析

第 2-1 章　基因组拼装与分析

第一节　基因组序列拼装概念

一、基因组短序列拼接问题

基因组测序与拼接是生物信息学一项重要任务。一个完整的基因组序列,对目标物种的遗传研究至关重要,特别是对该物种遗传育种、进化、基因功能等研究具有重要支撑作用。目前的高通量测序技术(包括第二和第三代测序技术)测序通量大,可以在短时间内获得目标物种基因组几十倍甚至几百倍基因组覆盖的 DNA 序列数据(详见第 1-1 章)。但是,第二代高通量测序仪产生的读序(read)一般在 100-150bp,对于一条染色体长度(几十 Mb 长度),如何利用这些短序列拼接出其基因组序列,对于生物信息学来说是一个挑战。简单来说,就是把基因组染色体序列打断,然后再拼起来(所谓鸟枪法测序,"shotgun sequencing")。我们不知道基因组整条序列是如何排列或组合的,同时,目前的技术又无法实现一次把整条染色体序列测通。所以,我们只有通过算法和计算机的帮助,把这些短的读序组装起来,成为一条完整的序列(所谓从头拼接,"*de novo* assembly")。

举个简单例子来说明短序列拼接问题:

例如我们有这样一句话:"It is just a hypothesis, so don't be seriously!",并把它写在一张纸上。假设我们现在不知道纸上这句话到底是什么,我们需想办法破解它。我们首先把这张纸复印很多份,并撕成碎片(当然还发生一件离奇的事:所有的空格和标点都消失了!)。我们从这些碎纸片中随机抽取,于是我们得到大量字条,上面分别有如下字母:

itis ypo stah the sodo eriou siss ju ntbes sly……

这就如同我们把基因组 DNA 打断,然后测定获得许多读序一样。我们不断抽取就不断得到更多不同类型的字母组合:

itis ypo stah the sodo eriou siss ju ntbes sly tis yopth sodon beser beser ssod iti sju……

另外,我们又发明了一种称作为"paired-end(PE)"或"mate-pair(MP)"的测序方法。这是通过构建不同插入长度测序文库来实现的。例如,构建片段长度 2Kb 的测序文库,高通量测序技术会测定出这条序列的两端各 100bp 左右的读序,而其中间约 1.8Kb 长度序列未知。但来自同一插入长片段的两端读序关系是已知的。我们的例子就像这样:

iti＊＊＊＊＊ahyp,sju＊＊＊＊＊pot,the＊＊＊＊＊don,sod＊＊＊＊＊ser bes＊＊＊＊＊sly,……

这样我们根据这些读序和它们的相关关系,我们可以把这句话拼回来:

"itisjustahypothesissodontbeseriously"

但它不是最终结果,我们根据我们的现有的语法习惯,我们给它们加上空格和标点,我们就能够还原原话!

第三代测序仪的出现,使测序的单个读序长度显著提高,一般平均读序长度可以达到5Kb以上,部分长读序可以达到20Kb甚至更长。但是目前三代测序的测序质量存在很大问题,错误率达到15%左右,这严重限制了这些序列的应用,包括基因组序列的拼接。为此,针对三代测序数据的特点,提出了大量基因组拼接的生物信息学拼接方法和策略(详见下面第三节)。

二、基因组从头拼接主要方法

就生物信息学而言,基因组序列拼接是指将测序获得的DNA片段,通过联配等手段组合出其原始序列。目前测序技术获得的读序长度从100bp到几十Kb不等,生物信息学必须提出相应的算法拼接出其原始序列。在基因组拼接过程中,往往可以利用遗传图谱等其他技术辅助拼接,使获得的拼接序列尽可能长,接近其原始序列长度。

基因组从头(*de novo*)拼接方法目前主要有两个主要方法,一个是基于读序之间的重叠序列(overlapped sequence)进行拼接的方法,所谓OLC方法(Overlap-layout-consensus),二是基于图论的算法。

OLC方法(图2-1.1)分三步,首先找读序可能的重叠区域(Overlap: find potentially overlapping reads),然后通过重叠区域拼接出序列片段(Layout: merge reads into contigs and contigs into supercontigs),最后基于片段关系和测序错误等信息确定最后的序列信息(Consensus: derive the DNA sequence and correct read errors)。

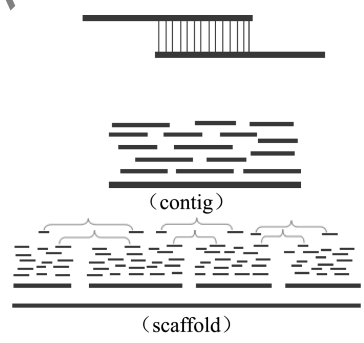


图2-1.1 序列拼接OLC(Overlap-layout-consensus)方法模式图

基于图论的基因组拼接算法是目前高通量测序读序拼接主流方法。OLC算法仅适用于第一代测序技术获得的读序(读序长度一般700bp左右)或一些大片段的拼接,但不适用于第二代测序的短读序(长度100~150bp)。对于短读序,由于重复序列等问题,OLC算法很难基于序列重叠获得一个正确的拼接结果,同时在实际运算过程中,大量重叠关系的读序信息需要大量内存,目前计算机能力难以承受。相反,基于图论的数据结构,特别适合处理大量具有重叠关系的短序列。下面一节将详细介绍该方法。

三、利用遗传图谱等进行基因组组装

对于基因组从头组装而言,Scaffold水平的基因组草图序列需要对大片段序列scaffold排序和定向,进行染色体水平的组装。染色体水平组装通常需要利用遗传图谱、光学图谱技术、Hi-C等等技术辅助组装。

1. 遗传图谱

利用遗传图谱(genetic map)辅助进行染色体水平组装又称之为"准染色体重建"(pseudo-chromosomes reconstruction),而其中'准'这个词代表的是基因组组装上仍旧存在很多不确定的地方,需要各种可能的证据来进行校验。目前遗传图谱中使用最广泛的为SNP标记。基于SNP标记进行连锁分析,构建高密度的遗传图谱。遗传图谱的标记、标记的遗传距离和标记对应的染色体是辅助基因组组装的重要信息。利用遗传图谱辅助进行遗传

图谱组装主要分“三步”:第一步,连接标记和 scaffold。通过全基因组序列比对软件 Blast 将一个个标记和 scaffold 连接起来,筛选标准一般为“Evalue 1e-6”,筛选结果取最佳匹配结果。每个标记因为其特异性的需要只能定位到一个 scaffold 上,而每个 scaffold 则需要尽可能多的标记来确定排序和方向;第二步,确定 scaffold 方向、排序和染色体定位。当定位到一个 scaffold 的标记数量大于等于 2 时,就有可能确定 scaffold 方向。通过定位到同一个 scaffold 的多个标记锚定到 scaffold 上的不同位置信息,并结合标记的遗传距离来进行 scaffold 方向确定。类似 scaffold 确定方向的方法,scaffold 排序的方法是通过标记对应的遗传距离和染色体信息来进行染色体定位;第三步,组装 scaffold 至染色体。当 scaffold 的排序、方向以及染色体定位这些信息已经清晰,最后一步是将 scaffold 一个个连接起来。目前通用的方法是 scaffold 之间用 100 个 N 连接起来。如果拼接的基因组已经有参考基因组,那么通过和参考基因组的序列共线性比较,就能判断‘准染色体重建’工作的优劣。如果没有参考基因组,可以通过和遗传图谱进行共线性的比较来判断是否存在大片段的漏洞(gap)。

当一个基因组存在多个可用的遗传图谱时,应尽可能多的使用遗传图谱,以增加标记数量和验证标记之间的一致性以及可能存在的图谱本身错误。标记数量的增加,一方面意味着可以更好地帮助 scaffold 进行排序和定向,另一面也有可能出现部分标记错位错误的问题。ALLMAPS(Tang 等,2015)是一个解决这些问题比较理想的生物信息学工具,它尽可能使输入序列和输入遗传图谱共线性保持一致,同时可以根据不同遗传图谱之间精确性等级,来设置权重大小来改善结果;它可以同时融合多个遗传图谱的软件,对于基因组染色体水平的组装有很大帮助。图 2-1.2 是软件 ALLMAPS 测试数据得到的染色体组装的可视化结果,图片主要展示了两个遗传图谱与染色体之间的共线性,当多个遗传图谱的重要性和准确性有区分时,可以通过权重(W)的改变进行染色体组装。例如异源四倍体榨菜基因组(*Brassica juncea*)使用 ALLMAPS 进行了染色体水平辅助组装(Yang 等,2016)。

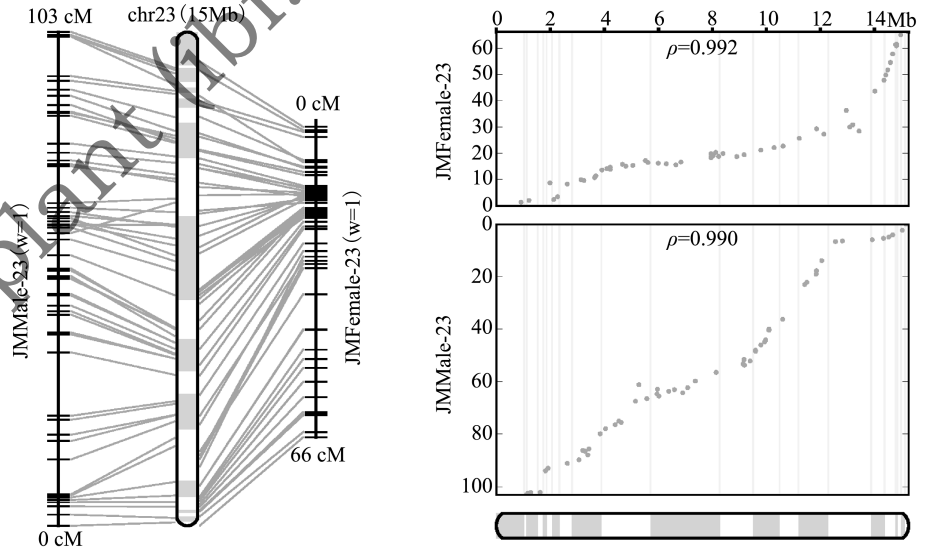


图 2-1.2　利用 ALLMAPS 辅助染色体组装的测试例举

2. 基因组组装新技术

利用遗传图谱组装基因组有其局限性,如需要构建遗传群体、周期长成本高等缺点。为此许多新技术不断被开发并应用于辅助基因组组装,光学图谱(BioNano)、Hi-C、芝加哥(Chicago)技术、10 X Genomics linked-reads 等各种技术应运而生。下面对这些技术进行简要介绍。

光学图谱技术利用酶切和荧光标记,对长达几百 Kb 的单链 DNA 分子进行成像,组装得到物理图谱。对于基因组组装,光学图谱能够提供大尺度下的远程信息,跨越重复片段和一些包含复杂元件的区域,可以用于辅助基因组复杂区域的组装。

Hi-C 技术指高通量染色体构像捕获技术(High-throughput chromosome conformation capture),是一种研究全基因组三维构象以及分析染色质片段相互作用的实验技术。对于基因组组装,Hi-C 技术主要是基于染色体内的相互作用远大于染色体之间的相互作用,近距离的相互作用大于远距离的相互作用,进行染色体基因组序列聚类、排序和定向到它们的正确位置,这与遗传图谱将基因组序列组装到染色体水平的方法类似。但遗传图谱需要构建专门的实验群体,而 Hi-C 技术只需要单个个体就可以实现序列染色体定位。

Chicago(体外 Hi-C)是一种优化 Hi-C 文库制备技术。由于 Hi-C 技术采用活细胞提取染色质方法构建大片段文库,因此,会存在一些生物学信号干扰,影响基因组组装的正确性。Chicago 技术以重组染色质为基础构建大片段文库,通过将 DNA、纯化的组蛋白以及染色质组装因子结合来重构染色质,去除生物学信号干扰。相对于 Hi-C,产出数据质量更高,组装准确性更好,但产出片段跨度范围相对小一些。

10 X Genomics linked-reads 技术本质上是在 DNA 片段上加入特异性条形码(barcode)序列,通过将 DNA 片段分配到不同的油滴微粒中,利用 GemCode 平台进行文库制备,并结合 Illumina HiSeq 测序平台进行测序。对于辅助基因组组装,该技术通过 barcode 序列信息追踪来自每个大片段 DNA 模版的多个读序,从而获得长片段序列的信息。基于长片段信息结合二代 Illumina short-Reads 组装的序列,可进一步对基因组进行组装,从而提升 scaffolds 长度。

第二节 基于图论的拼接算法

一、图论

说到图论(graph theory),就不得不先说大数学家欧拉(Leonhard Euler,1707—1783)和哥尼斯堡的七桥问题。有条河穿过哥尼斯堡(Konigsberg,现俄罗斯加里宁格市),形成 2 个大岛,市内建有 7 座桥连接这些岛(图 2-1.3)。哥尼斯堡的市民始终在想一个问题:是否可能从一个地点出发,经过 7 座桥且每座桥只过一次,然后回到出发地? 1735 年,欧拉给出了答案:不可能。欧拉把这个问题抽象成一个由点和线构成的图问题(图 2-1.3):可能的出发地为 4 个点(A\B\C\D),7 座桥为 7 条线,这样市民的问题变成从任何一个点出发,经过 7 条线且仅路过一次,再回到原点。由于该图不存在欧拉回路(Eulerian circuit)(详见下面说明),所以不可能找到这样一条路径满足上述要求。欧拉对这个问题的抽象及其解决算法被认为是图论学科的起始。

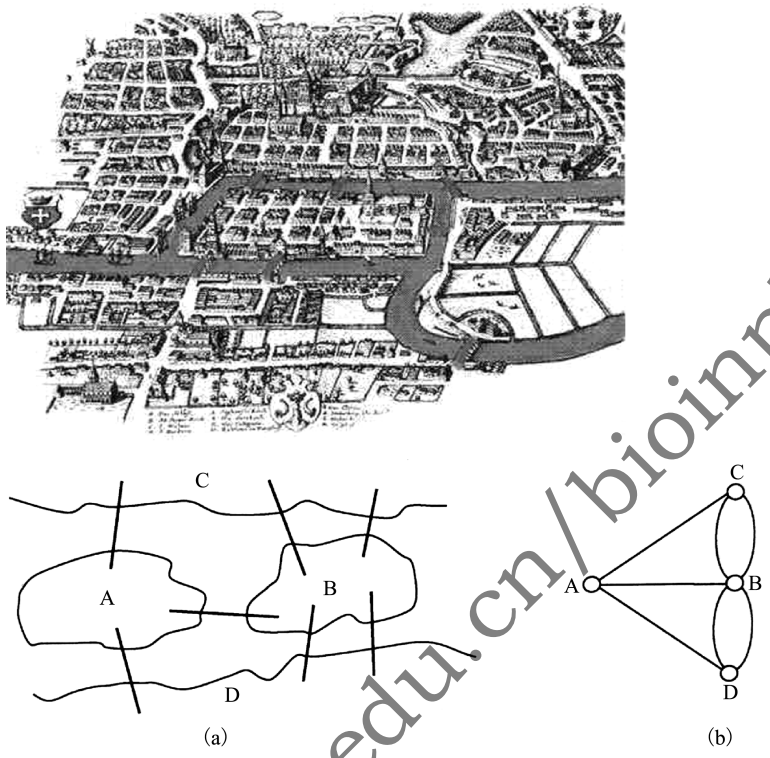


图 2-1.3　哥尼斯堡七桥问题及其抽象图解

1. 图的基本术语及最短路径问题

图论中涉及的图由顶点(vertex)、边(edge)和关联函数(incident function)组成。关联函数是指使一张图中每条边对应于顶点的规则;顶点的度(degree)是指作为边的端点的个数;边分有向和无向边,边的权重(weight)指边的长度等等。

路径(walk)是指一张图的一部分,顶点和边交替连接。途径允许重复经历点和边。同一图中各边互不相同的途径叫迹(tail),起点和终点相同的迹叫回(circuit)或回路;同一图中各顶点互不相同的途径成为路(path),起点和终点相同的路叫圈(cycle)。连通图(connected graph)是指同一图中任何两个顶点都是连通的(图 2-1.4)。

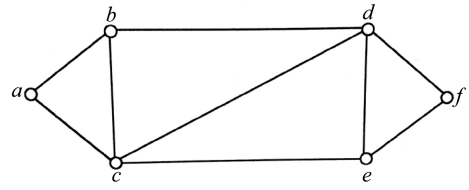


图 2-1.4　一个具有 6 个顶点的连通图

一个边设有权重的图中,对于任何连接两个顶点之间的边,其权重合计最小的路径叫作这两个顶点间的最短路径。在实际许多具体问题,都可以转化为寻找最短路径问题,如两条序列联配寻找最优联配问题。如何确定图中最短路径?荷兰计算机科学家 Dijkstra(1959 年)提出了一个寻找联通图最短路径的有效算法(1972 年获得计算机领域大奖图灵奖)。该算法类似动态规划算法,这里就不进行讲解。

2. 欧拉图与哈密顿图

欧拉图(Euler graph)是指含有欧拉回路的连通图。例如图 2-1.4,其为欧拉图,其中

“*bacbdcedfeb*”就是一个欧拉回路。经过图中所有边的迹叫欧拉迹,欧拉回路就是起点和终点为同一顶点的欧拉迹。判断一个联通图是否为欧拉图,可以根据其每个顶点的度是否都是偶数,欧拉图顶点特征就是具有偶数度。对于一个欧拉图,找到其欧拉回路并非易事。Flewry 于 1921 年提出了一个在欧拉图中寻找欧拉回路的算法。同时,也有一些方法可以计算一个欧拉图中不同欧拉回路的数量。

哈密顿图(Hamilton graph)是指含有哈密顿圈的图。哈密顿圈指经过图中所有顶点且起点和终点为同一顶点的圈,例如图 2-1.3 中“*abdfeca*”就是一个哈密顿圈。哈密顿路径是指经过图中所有顶点的路径。判断一个图是否是哈密顿图是一个 NP-完全问题;同样,在一个图中寻找哈密顿路径也同样不容易,也没有有效算法解决。

二、基于德布鲁因图的拼接算法

1. 德布鲁因(De Bruijn)图

要说清楚德布鲁因图,就必须先说德布鲁因序列(De Bruijn sequence),它是德布鲁因在数学领域最重要的一个贡献。德布鲁因序列,$B(k, n)$,是指 k 元素构成的循环序列。所有长度为 n 的 k 元素构成的序列,都是它的子序列,出现并且仅出现一次。

例如,二进制序列“00010111”,属于 $B(2,3)$ 序列,即 2 个不同元素(0,1);所有长度为 3 的子序列或子串为 000,001,010,101,011,111,110 和 100,正好构成了 2 个元素长度为 3 的所有组合。也就是说,德布鲁因序列对于每个子序列是递进的,环环相扣,循环往复。

德布鲁因序列可以通过确定 n 维德布鲁因图的哈密顿路径,或 $n-1$ 维德布鲁因图的欧拉路径进行构建。以图 2-1.5 为例,构建其 $B(2,4)$ 序列。这是一个 3 维德布鲁因图,顶点由 3 个数字组成的子序列,边为 4 个数字组成。如果一条路径恰好通过每条边一次并且回到起始点,那么每四个数的亚序列(图的边)会出现正好一次(该路径为欧拉回路);如果路径刚好经过每个顶点一次,那么每三个字的亚序列(图的顶点)出现正好一次(该路径为汉密顿路径)。假如我们沿着如下欧拉路径行走:

000, 000, 001, 011, 111, 111, 110, 101, 011, 110, 100, 001, 010, 101, 010, 100, 000

这样就形成如下 $k=4$ 的子序列串:

0000
_0001
__0011
……

对应的德布鲁因序列为“0000111101100101”

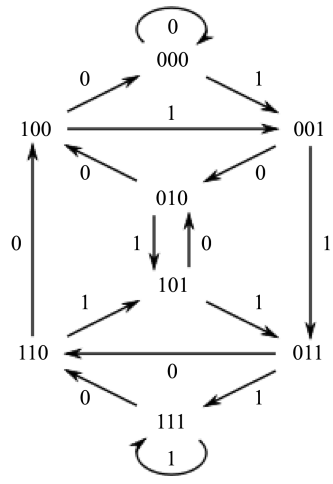


图 2-1.5 德布鲁因(De Bruijn)路径列举

该图为 3 维德布鲁因图(顶点由 3 个数字组成,边为 4 个数字组成)。

2. 基于德布鲁因图的基因组拼接算法

如上所述,基因组拼接主要利用两类算法,一是 OLC 算法,适用于传统桑格测序数据,而基于德布鲁因图的算法,是目前用于第二代高通量短序列数据拼接的主要算法。基于德

布鲁因图的数据结构,特别适合处理大量具有重叠关系的短读序:该数据结构中,利用读序 *K*-mer 作为顶点,读序作为边,这样总体上说,图的大小就是由目标基因组大小和重复序列含量决定,而与读序覆盖深度无关(Li 等, 2010)。这里 *K*-mer 是指一条字符串中所有可能具有长度为 k 的子串。在计算基因组学中,*K*-mer 是指一条测序获得的序列中,所有可能具有长度为 k 的子序列或亚序列。对于一条长度为 L 的序列,所有可能的长度为 k 的子序列数量为 $L-k+1$,而可能的 *K*-mer 数量与序列构成元素数量(n)有关(如 DNA 序列由 4 个碱基构成,$n=4$),所有可能数量为 n^k 个。

俄罗斯生物信息学家帕夫纳(Pavel A. Pevzner)第一次把德布鲁因图引入序列的拼接。1989 年,他首次应用德布鲁因图用于杂交测序技术(Sequencing by hybridization, SBH)的序列拼接问题,随后他和美国生物信息学家沃特曼(Micheal S. Waterman)一起,正式引入序列拼接并开发了拼拼软件用于实际序列拼接(Pevzner 等, 2001)。他们提出在德布鲁因图中,可以通过寻找欧拉路径的思路来确定拼接序列(即德布鲁因序列)。随着高通量测序技术的出现,该拼接算法成为基于高通量数据的基因组拼接主流方法。

在德布鲁因图中寻找欧拉路径是非 NP 问题,已有不少算法解决这个问题。例如 Flewry1921 年就提出从一个欧拉图中找欧拉回路的算法。

根据一条读序可以确定其 *K*-mer 及其以各个 *K*-mer 为顶点的德布鲁因图。以读序“AGATACT”为例,其 3-mer($K=3$)及其构成的德布鲁因图如下:

Read:	A	G	A	T	A	C	T
	A	G	A				
		G	A	T			
3-mer:			A	T	A		
				T	A	C	
					A	C	T

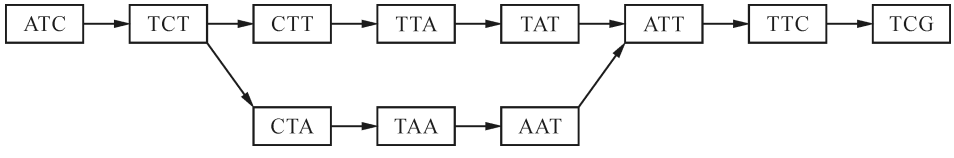
AGA → GAT → ATA → TAC → ACT

基因组高通量测序会产生许多读序,会存在杂合性、重复序列和测序误差等等问题,由此构成的德布鲁因图会很复杂。

例如,如果两条读序只有一个碱基差异(杂合性或测序误差造成),它们的德布鲁因图就会形成一个小包(bubble):

ATCTTATTCG

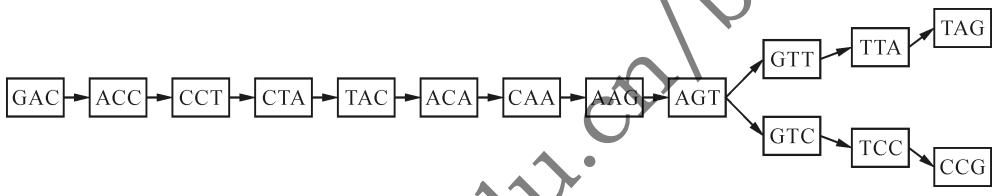
ATCTAATTCG



如果 10 条读序有如下重叠关系：

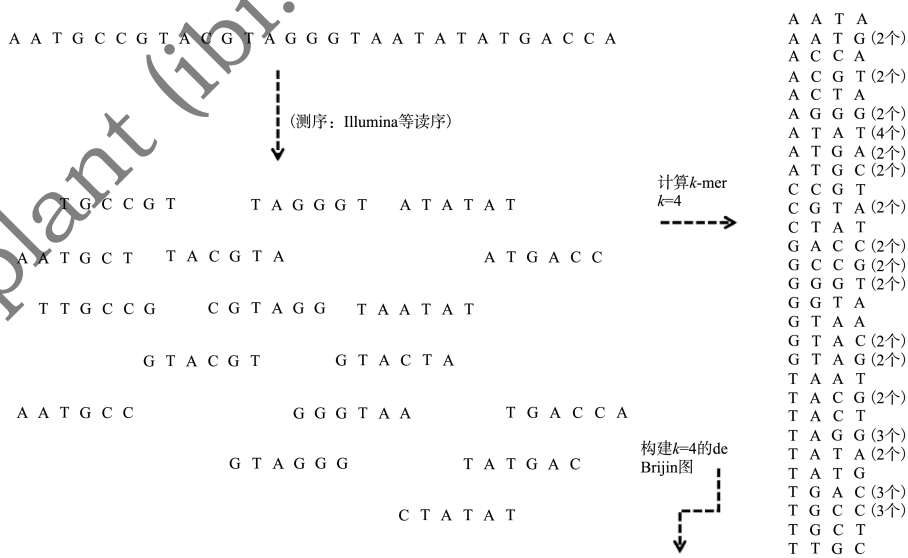
```
G A C C T A C A
  A C C T A C A A
    C C T A C A A G
      C T A C A A G T
        T A C A A G T T
          A C A A G T T A
            C A A G T T A G
        T A C A A G T C
          A C A A G T C C
            C A A G T C C G
```

基于这 10 条读序构建的德布鲁因图会形成叉型(重复序列导致)：



一个更复杂的例子(引自 Velvet)如下：

来自一条基因组序列片段(绿色序列),打断测序产生的读序为 6nt 长度(读序中红色碱基为测序误差)。我们确定其 4-mer 及其覆盖度(出现个数),可见部分 4-mer 出现 3-4 次,但含有特定测序误差的 4-mer 出现次数都仅为 1 次。基于这些 4-mer 构建德布鲁因图(图 2-1.5)。图中由于测序误差产生了错误分叉和小包。



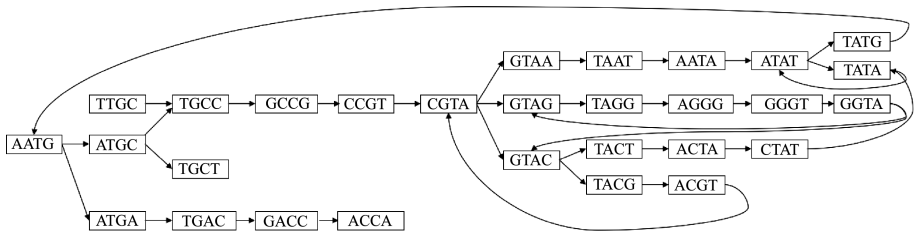


图 2-1.5　基于基因组序列读序构建德布鲁因图列举。基于来自一段基因组序列片段的 6nt 长度测序读序(读序中红色碱基为测序误差),获得其 4-mer 及其德布鲁因图(引自 Velvet)。

如果把来自一个基因组测得的所有读序,放到一起构建德布鲁因图,可以想象这个图将非常复杂和庞大,测序误差、重复序列等会使图产生大量错误连接。所以,一般在进行基因组拼接前,需要首先对测序读序进行错误纠正或质量控制,去除图中可能的错误分叉或小包等。

以华大基因研发的基因组拼接软件 SOAPdenovo 为例进行具体说明。

图 2-1.6 大致给出了 SOAPdenovo 的拼接过程:基因组序列打碎到一定长度构建测序文库(图中表示不同长度的库,短的 150~500bp 和长的 2~10kb),进行双端(PE)高通量测序,每条读序长度约 100bp,然后将获得的大量读序构建 *K*-mer(*K* 大小根据每个物种进行测试,选择拼接效果理想的长度,如 40~50bp)。以这些 *K*-mer 为顶点,根据 *K*-mer 关系(包括 PE 关系)构建德布鲁因图。然后对初步构建而成的图进行修正,去掉由于测序误差等引起的错误连接。根据最后确定的德布鲁因图,寻找欧拉路径,即确定可能的拼接序列。一般是基于 contig 进行两头延伸;对于大的重复序列,由于形成剪刀叉形图,拼接无法确定两端连接方式,一般做法是在拼接序列中去除这段重复序列,然后根据 PE 关系,确定最后的拼接序列(scaffold)。然后再将读序定位到拼接的序列上,根据定位的读序可以将拼接序列的缺口(gap)大小尽量缩小。这样一个完整的拼接过程就结束了。

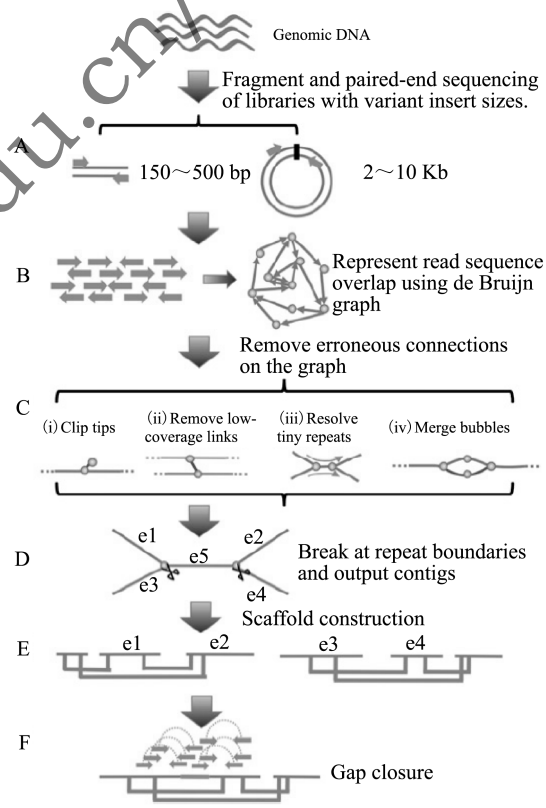


图 2-1.6　基于德布鲁因图的基因组拼接算法例举(以 SOAPdenovo 为例)(引自 Li 等, 2010)

SOAP(Short Oligonucleotide Analysis Package)系列是华大基因针对二代测序数据(NGS)自主开发的分析软件包,其中 SOAPdenovo 是应用较为成功的拼接软件。SOAPdenovo 是为动植物大基因组设计的基于短序列的从头拼接软件,同时其在较小基因组(细菌和真菌基因组)的拼接中也可以应用。目前 SOAPdenovo 共有两个版本,分别于 2010 年和 2012 年发表

在 *Genome Research* 和 *GigaScience* 杂志上。相对于 SOAPdenovo 而言,SOAPdenovo2 通过更新算法设计,降低了内存消耗,解决了 contig 拼接中重复区域对于拼接的影响,提升了 scaffold 构建中长度和覆盖度,以及进一步优化了补洞程序,拼接效果有了非常明显的提高。相较于 Velvet, Allpath-LG 等拼接软件,SOAPdenovo2 的安装和程序运行十分方便。在程序运行方面,SOAPdenovo2 可以仅仅用一个命令行(SOAPdenovo-127mer all -s xxx.config -K 31 -d 1 -F -o output)即可完成拼接任务。在这个命令中"all"代表一次性运行所有拼接任务。"-K"后的数字是拼接所用的 *K*-mer 长度(奇数),通常需要尝试很多值,选取最佳拼接效果的 *K*-mer 长度。"-d"后面的数字指保留大于该数字频率的 *K*-mer 用于拼接,该参数目的在于去除测序错误导致的低频 *K*-mer 对拼接的影响。"-F"是利用读序对 scaffold 进行补洞。"-s"后是指拼接所需的配置文件。

具体配置文件相关信息如下:

```
##配置文件
# [LIB]
#文库信息以此开头
avg_ins=200
#average insert size (文库插入片段长度)

reverse_seq=0
#if sequence needs to be reversed(序列是否需要被反转,目前的测序技术,插入片段大于等于 2k 的采用了环化,所以对于插入长度大于等于 2k 文库,序列需要反转,reverse_seq=1,小片段设为 0)

asm_flags=3
#in which part(s) the reads are used
#该文库中的 read 序列在组装的哪些过程(contig/scaff/fill)中用到
#设为 1:只用于构建 contig;
#设为 2:只用于构建 scaffold;
#设为 3:同时用于构建 contig 和 scaffold;
#设为 4:只用于补洞、
#短插入片段(<2K)的设为 3,同时用于构建 contig 和 scaffold,长插入片段(>=2k)设为 2,不用于构建 contig,只用于构建 scaffold,454single 长 reads 只用于补洞。

rd_len_cutoff=100
#use only first 100 bps of each read
#默认选取读序前 100bp 序列用于后续分析,因为一般测序读序尾部质量相对较低,该值一般设置的比实际读序长度稍短一些,具体长度可根据每个库序列质量报告。

rank=1
#in which order the reads are used while scaffolding
```

#rank 该值取整数,决定了 reads 用于构建 scaffold 的次序,值越低,数据越优先用于构建 scaffold。设置了同样 rank 的文库数据会同时用于组装 scaffold。一般将短插入片段设为 1;2k 设为 2;5k 设为 3;10k 设为 4;当某个档的数据量较大时,也可以将其分为多个档,同样,当某档数据量不足够时,可以将多个档的数据合在一起构建 scaffold。这里说的数据量够与不够是从该档的测序覆盖度和物理覆盖度两个方面来考虑的。

pair_num_cutoff=3

cutoff of pair number for a reliable connection (at least 3 for short insert size)

#可选参数,pair_num_cutoff 该参数规定了连接两个 contig 或者是 pre-scaffold 的可信连接的阈值,即,当连接数大于该值,连接才算有效。短插入片段(<2k)默认值为 3,长插入长度序列默认值为 5

map_len=32

#minimum aligned length to contigs for a reliable read location (at least 32 for short insert size)

#map_len 该参数规定了在 map 过程中 reads 和 contig 的比对长度必须达到该值(比对不容 mismacth 和 gap),该比对才能作为一个可信的比对。可选参数,短插入片段(<2k)一般设置为 32,长插入片段设置为 35,默认值是 K+2。

q1=/path/**LIBNAMEA**/fastq1_read_1.fq

q2=/path/**LIBNAMEA**/fastq1_read_2.fq

#a pair of fastq file, read 1 file should always be followed by read 2 file

双端测序的一对 fastq 文件,/path/**LIBNAMEA**/为文件储存绝对路径

###由于篇幅有限,其他类型的序列输入文件格式就不再做介绍,详情可参考 https://github.com/aquaskyline/SOAPdenovo2

第三节　基于第三代测序数据的拼接

目前第三代测序技术主要包括 Helicos Bioscience 公司的 tSMS、Pacific Biosciences 公司的 SMRT 以及 Oxford Nanopore Technologies 公司的 Nanopore sequencing 技术(详见第 1-1 章)。由于测序深度高度不均一、测序错误较高以及融合读序(chimeric read),造成三代单分子测序数据用于拼接组装非常具有挑战性。目前 PacBio(PB)的 SMRT 技术是应用较为广泛的三代技术,本节就以 PB 数据为例,对三代测序数据拼接及相关软件进行概述。

根据三代测序深度的不同,可以考虑不同的拼接方式。大致有三种策略(图 2-1.7):(1)完全利用三代 PB 数据从头拼接(所谓“PB-only”或“Non-hybrid assembly”),这需要至少 50×的测序覆盖深度,现有的一些算法如 HGAP(hierarchical genome-assembly process)等在至少 50×的测序覆盖深度才能发挥出最好的拼接效果,有时为了更好的拼接效果,可能会要求 80×甚至 100×的测序覆盖深度;(2)二代和三代数据混拼(所谓“hybrid”),建议 5-50×

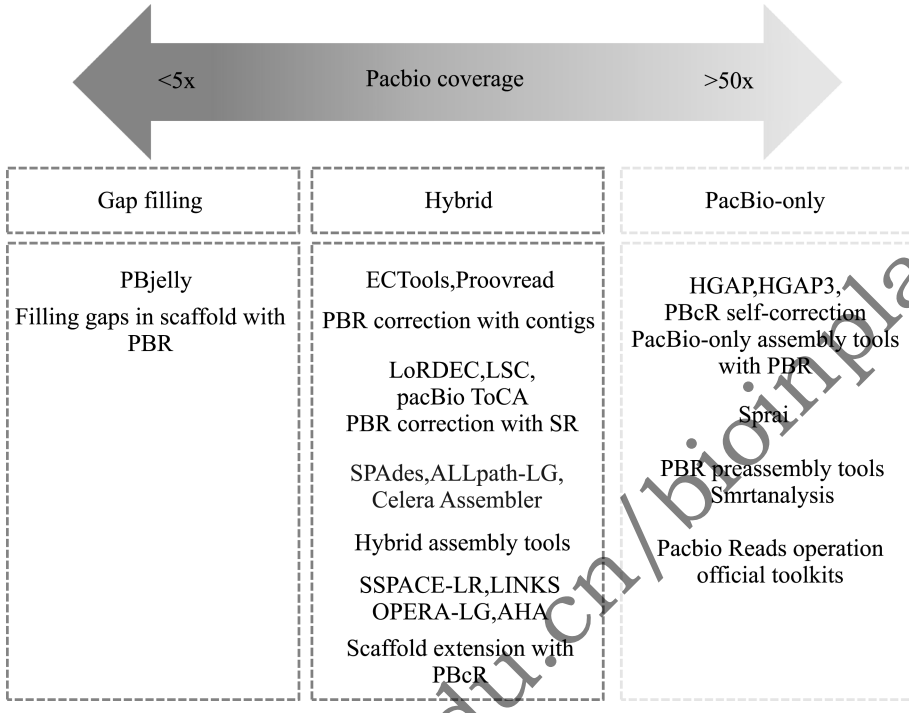


图 2-1.7 根据 PacBio 测序覆盖深度不同,采取不同的序列拼接策略(改自 Schatz,2014)

的三代测序覆盖深度的数据,目前主流算法如 PBcR 和 ECTools 等,它们在 20×左右的测序覆盖深度或以上,可以获得不错的拼接效果。三种主要拼接策略中,二和三代数据混拼相对比较复杂,涉及长短类型读序,图 2-1.8 进一步给出了拼接过程中各结果之间的关系和相应拼接流程;(3)三代数据仅用于二代数据拼接结果的补洞(gap filling),这种策略建议在已有高质量的二代拼接结果(scaffold 水平)的情况下,产生 5×左右的三代测序覆盖深度,利用 PBjelly 算法进行补洞改善拼接。

进一步对上述三种拼接方法涉及的具体拼接软件进行了汇总(表 2-1.1)。

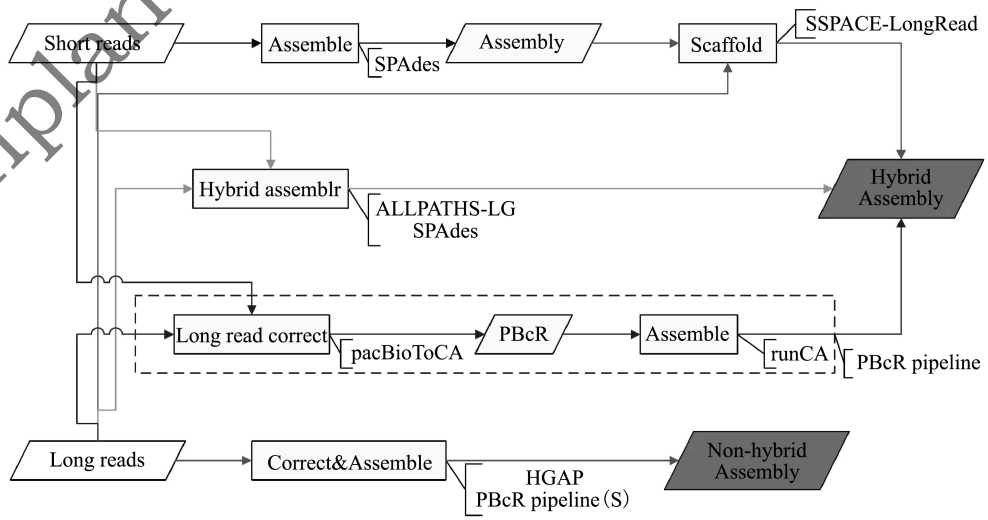


图 2-1.8 二代短读序(Short read)和三代长读序(Long read)数据混合拼接的流程图(改自 Liao 等,2015)

表 2-1.1　三代 PacBio(PB)数据三种拼接策略涉及的算法软件汇总 *

拼接策略	软件名称	描述
完全基于 PB 数据	HGAP	完全基于 Pacbio 序列拼接流程,基因组拼接上限目前为 130Mb,如需更大基因组可尝试 smrtmake 中的 HGAP3
	Falcon	一个二倍体三代拼接软件
	PBcR self-correction	与 HGAP 原理相同,以 PBcR 为基础进行三代序列自我修正的流程
	Celera® Assembler	可供 PacBio 序列直接拼接的软件
	Sprai	以产生更长 contig 为目的的拼接软件
二代和三代数据混用	pacBioToCA	Celera Assembler 中的一个模块,利用二代序列联配到三代序列上,以达到修正三代序列并产生一致序列。
	ECTools	一个利用二代拼接的 contig 修正三代序列的软件包
	SPAdes	利用二代和三代序列进行混合拼接的软件
	Cerulean	基于 ABYSS 算法利用二代序列来延长 contig 并解决其中的拼接小包(bubble)
	dbg2olc	利用二代 contig 与三代序列混合拼接的软件
	proovread/	利用二代 contig 对三代序列进行大规模高精度纠正
仅用于二代补洞	PBJelly	通过三代数据来填补二代数据拼接获得 scaffold 中的洞,来改善拼接质量。应用的基因组大小可以超过 1G

* 来自 https://github.com/PacificBiosciences/

下面具体说明三种拼接策略流程:

1. 基于 PacBio 测序数据从头拼接

仅用 PacBio 的测序数据进行从头(*de novo*)组装。三代数据在进行组装之前需要进行预处理,降低三代数据中的错误率,利于后续拼接。利用的拼接算法是 OLC 算法。三代从头组装目前最主要软件工具包括 HGAP 和 PBcR (PacBio corrected reads pipeline via self-correction)。目前 HGAP 能够拼接的基因组上限为 130MB,不适合较大基因组的拼接。HGAP 整个拼接流程共分为三个步骤:预组装(preassembly),组装(assembly)和组装后修饰(consensus polishing)。

预组装:整个流程中,预组装是 HGAP 是否能够成功组装至关重要的一步,这一步的目标是产生比初始序列更长更加准确的序列。通过将三代数据中的短序列比对到长序列(seed reads)上,将产生的一致序列(consensus sequence)进行质量控制和序列修整,最终实现预组装。

组装:对于读序长度平均达到 5-10Kb 的 Pacbio 数据而言,选择 OLC 算法显然是比较明智的。组装成功的关键在于整体的序列覆盖深度,预组装序列的长度分布以及整个基因组的重复序列比例也很关键,对于基因组的大跨度重复区域,必须有足够长足够覆盖度的读序才能够组装成功。

组装后修饰:PacBio 组装草图中仍然存在着许多插入删除(Indel)错误和碱基替换

(SNP)错误。修饰这些错误需要每个 Pacbio 序列文件所附带的一个后缀为“bas.h5”的文件,里面包含着每条序列插入、删除、替换以及融合碱基的质量值,代表着错误的可能性,而 Quiver 的算法则是通过“bas.h5”文件来计算出一致性好质量高的最终拼接版本。

2. 二代三代数据混合拼接

利用三代 PacBio 数据于二代的读序数据或者 contig 数据进行混合组装。

二代和三代数据混合拼接可以用 PacBioToCA 和 ECTools 等工具(图 2-1.7)。两者的共同点均是利用二代数据对三代序列进行修正,最终的拼接都是由软件 CA(Celera Assemble)完成,不同的是 PacBioToCA 本身为 CA 的一个模块,利用二代读序对三代序列进行修正,而 ECTools 则是利用二代预拼接好的 contig(CA 软件推荐为 unitig)进行修正。也可以利用 ALLLPATH-LG 和 Spades 工具,对三代序列不做修正,通过设定特定参数来与二代读序进行混合拼接。

3. 三代数据用于补洞改善拼接

利用 PacBio 数据可以对基于二代数据拼接成的 scaffold 中的“N”进行填补。目前将三代数据用于二代 scaffolds 补洞主流的软件有 PBjelly 等。值得一提的是,PBjelly 并不仅仅针对三代数据,也可以用于其他长序列,如 454 测序平台的读序。该工具通过将长序列联配到高质量基因组拼接草图上,发现并修正基于二代数据拼成的 scaffold 序列上的缺口,达到提高二代数据草图质量的目的。

第四节 基于字符串(*K*-mer)的基因组调查与分析

在正式启动一个基因组测序项目前,往往需要首先对目标物种进行所谓基因组调查(genome survey)测序和分析。该测序一般进行短片段插入库 30-70×基因组覆盖测序,并利用该数据进行 *K*-mer 分析,获得目标物种基因组的大小、杂合度、倍性和重复序列比例等基本参数的估计。该估计结果对目标基因组的拼接和分析具有重要指导意义。

如上所述,*K*-mer 是指一条字符串中所有可能具有长度为 k 的子串(子序列)。对于一条长度为 L 的序列,所有可能的长度为 k 的子序列数量为 $L-k+1$。*K*-mer 在生物信息学分析中用途非常广泛,除了上述 *K*-mer 被用于基因组序列拼接外,它同时也被用于基因组大小估计、杂合度、重复序列估计等。

一、基因组大小估计

基于测序结果,假设我们获得了一个基因组的所有 *K*-mer,根据 Lander-Waterman 模型(1988),基因组大小(G)可以根据如下公式估计:

$$G = K_{num} / K_{depth}$$

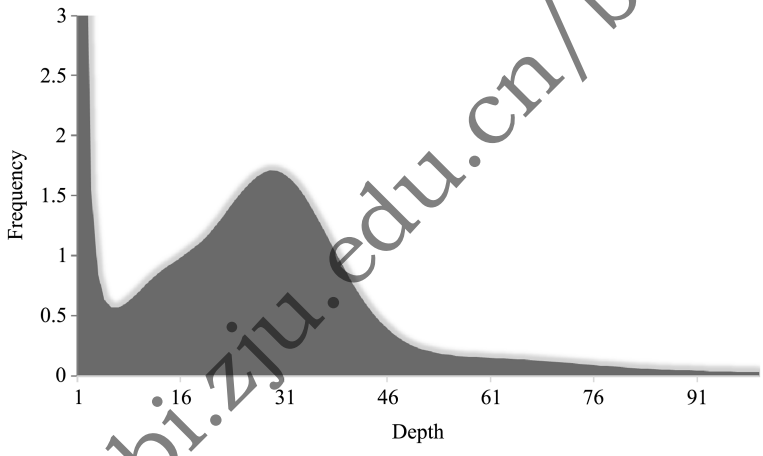
式中 K_{num} 是 *K*-mer 的总数, K_{depth} 是 *K*-mer 的期望测序深度。

人类基因组测序工程初期,需要构建基因组的物理图谱。物理图谱构建过程中,一个棘手问题是需要挑选多少克隆才能覆盖整个基因组?挑选太多,工作量巨大,太少无法覆盖整个基因组,物理图谱质量不高,无法完成基因组测序。为此,Lander 和 Waterman(1988)进行

了理论测算,提出了上述方法并给出了一些参数的统计特征。后来,Li 和和 Waterman(2003)等把它引入基因组调查序列数据,特别是高通量基因组测序数据为基础的基因组大小估计。

K-mer 的总数可以根据获得的所有读序进行估计。如果我们能进一步知道 *K*-mer 的期望测序深度,我们就可以基于上述公式估算出基因组大小。根据 Lander 和 Waterman(1988)分析,*K*-mer 深度频率分布遵循泊松分布,我们可以根据 *K*-mer 频率分布曲线的峰值作为其期望深度。

以 17-mer 为例,估计一个禾本科物种菰草(*Zizania latifolia*)的基因组大小(Guo 等,2015)。我们首先构建了一个短序列测序库并测定了大约 35×基因组覆盖度序列,基于该数据我们得到约 1.72 亿个 17-mer,其深度分布图如图 2-1.9。根据此图,可见其 17-mer 深度分布峰值在 29×处。由此,我们估计其基因组大小为 *k*-mer 数量/ *k*-mer 深度 = 17238.2/29 = 594.4Mb。该结果与流式细胞仪测定的该物质基因组大小一致。



样品	*K* 值	*K*-mer 数量	深度	基因组大小	有效碱基数	read 数量	测序深度
菰“HSD02”	17	17 238 224 304	29	594 421 527	20 521 695 600	205 216 956	34.5

图 2-1.9　禾本科物种菰草(*Z.latifolia*)基因组调查测序及其 17-mer 深度分布图(引自 Guo 等,2015)

二、基因组复杂度估计

由于杂合性和倍性等因素,使一些物种基因组变得异常复杂,增加了基因组拼接的难度。*K*-mer 频率分布曲线会由于基因组杂合性、倍性、重复序列等因素发生变化,这些变化为我们提供了目标基因组非常有用的信息。

基因组的杂合性,会使来自杂合区段的 *k*-mer 深度较纯合区段降低 50%。例如,来自基因组的一个 17-mer 片段,如果没有杂合性,其深度为 2;如果有一个杂合位点,则这个片段将会有 2 个 17-mer,同等测序量情况下,2 个 17-mer 的厚度均为 1。因此,如果目标基因组有一定的杂合性,会在 *k*-mer 深度分布曲线主峰位置(c)的 1/2 处(c/2)出现一个小峰(图 2-1.10A)。同时,杂合率越高,该峰越明显。如果目标基因组为多倍体物种,特别是同源多倍体或相近物种杂交形成的多倍体,两个或多个基因组序列高度同源,许多长序列片段(>*k*-mer 长度)甚至完全相同,在测序量一定的情况下,这样就导致相应区域的 *k*-mer 数量会成

倍性增加,在 *k*-mer 深度分布曲线上就会在主峰深度位置的 1 倍(4 倍体)或 1 和 2 倍处(6 倍体)出现 1 个或 2 个峰值。以一个禾本科六倍体物种稗草(*Echinochloa crus-galli*)为例,我们构建了一个短序列测序库并测定了其 40×基因组序列,基于该数据构建了其 17-mer 深度分布图。从该图可见 3 个明显的分布峰(图 2-1.10B)。如果基因组重复序列很高,导致高深度的 *K*-mer 数量增加,其 *K*-mer 深度频率分布右端会出现一个比较明显的拖尾(图 2-1.10A)。一个武断但基本靠谱的重复序列比例估计方法,就是将大于 2c 深度的 *K*-mer 在调查测序数据集中的比例,作为目标基因组重复序列的估计值。

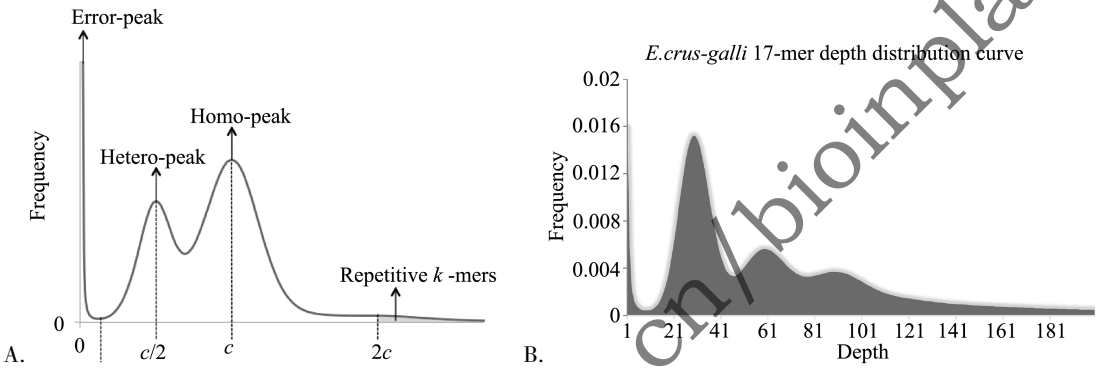


图 2-1.10 复杂基因组 *K*-mer 深度分布图及其特征峰

A、一个具有复杂基因组 *K*-mer 深度分布模式图。基因组杂合性和重复序列会在 *K*-mer 深度分布产生特征峰;B、多倍化基因组的 *K*-mer 深度分布图。以禾本科六倍体物种稗草(*E. crus-galli*)为例。

如果读序测序质量不高,导致出现大量碱基测序误差,这样会使低深度(1-2×)的 *K*-mer 数量大量增加,而使主峰不明显或不出现,同时如果测序深度不够(特别是目标基因组比较大情况下),也同样无法使目标基因组的主要 *K*-mer 分布特征出现。基因组 DNA 的 *K*-mer 分布特征会随基因组的复杂性增加而变化,一些更为复杂的基因组 *K*-mer 分布特征可参见一些理论研究结果(如 Chor 等,2009)。由此可见,基因组的 *K*-mer 分布,就像我们体检中的很多生化指标一样,可以为我们了解基因组基本状况发挥重要作用。

三、基因组“肖像”及缺失字符串分析

对于一个基因组给定的 *K*-mer,就有 4^K 种不同的字符串,为了反映每种字符串的数量,可以把每个字符串数量排在一个 $2^K \times 2^K$ 的方块中(如图 2-1.11 上),这样的方阵成为 *K* 框架。对于不同大小的基因组和 *K*-mer,可以使用同样大小的 *K* 框架。由于一个特定 *K* 串的计数可能为 0(缺失)或某一整数,可以采用一个粗略的颜色标尺来反映计数结果,如白色表示缺失,鲜艳的颜色表示计数比较小,而深颜色(到黑色)表示计数比较大。这样用颜色表示某一物种基因组 *K* 框架就得到了该物种的一个“肖像”。不同物种其基因组“肖像”不同。郝柏林院士实验室提出了上述方法并致力于细菌基因组“肖像”研究。利用他们编写的生物信息学工具 SeeDNA, 可以获得大肠杆菌 K12 菌株的基因组“肖像”(*K*=8)(图 2-1.11)。

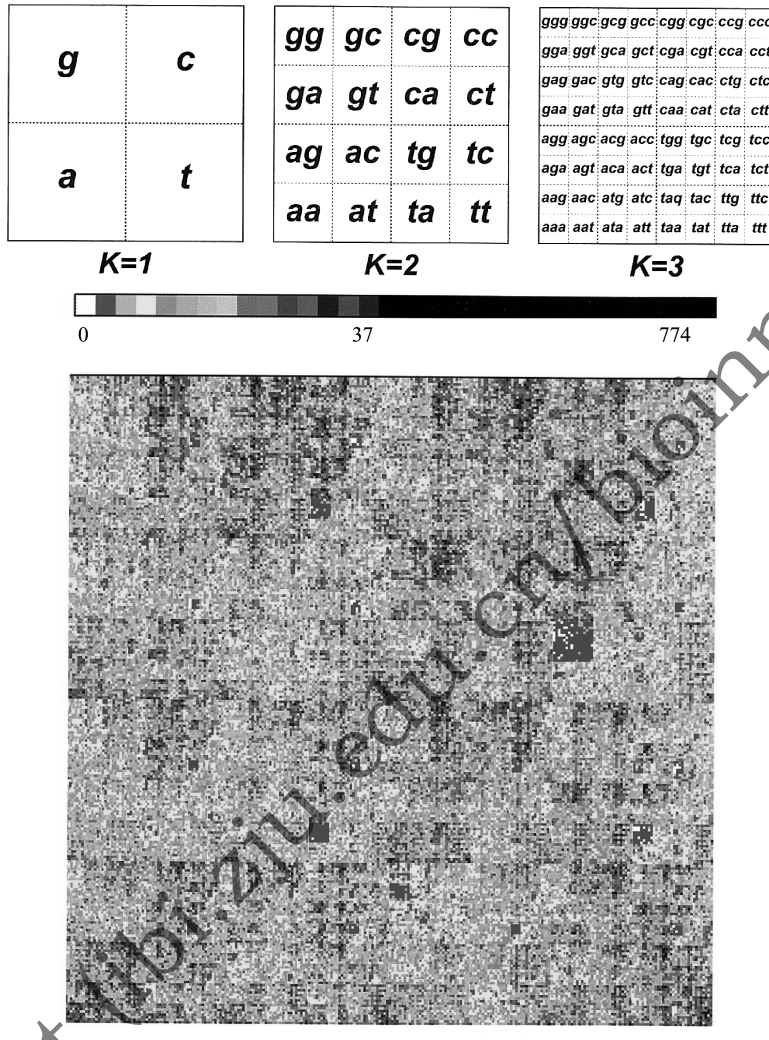


图 2-1.11　基因组 *K* 框架(上图)及其细菌基因组“肖像”(引自郝柏林,2015)

基因组“肖像”研究可以发现基因组中一些特定 *K* 串特别稀少,甚至缺失,如大肠杆菌基因组中“ctag”就特别稀少,这是生物学家已经知道的事实。郝柏林院士实验室详细分析了各种细菌的基因组“肖像”,发现了一批细菌基因组显著缺少的字符串(表 2-1.2 列出了 *K*=4 的结果)。

表 2-1.2　一批细菌基因组缺失或显著缺少的四字母(*K*=4)字符串(引自郝柏林,2015)

菌种\字符串	ctag	acgt	gatc	gtac	tcga	gcgc	cgcg	ggcc	ccgg
大肠杆菌	√								
沙门氏菌	√								
志贺是痢疾杆菌	√								
海栖热袍菌	√								

续表

菌种\字符串	ctag	acgt	gatc	gtac	tcga	gcgc	cgcg	ggcc	ccgg
耐辐射奇球菌	√								
根瘤菌	√								
枯草芽孢杆菌	√								
产甲烷热自养古菌	√					√	√		
苍白密螺旋体	√							√	
产水菌	√				√	√		√	
詹氏支原体	√		√	√		√	√		
肺炎支原体									√
幽门螺旋菌		√		√	√				
流感嗜血菌								√	√
伯氏疏螺旋体							√		
集胞菌						√	√		
强烈炽热球菌						√	√		

上述缺失的字符串序列全为回文结构(即第 1 和 4 个字母满足 Crick-Watson 配对,第 2 和 3 字母配对)。缺失的字符串多为回文这一事实,说明它们与限制性内切酶识别位点有关。上世纪 70 年代,限制性内切酶首先在细菌中发现,作为其防御的一个武器,细菌利用该酶可以剪切外来 DNA 序列。同时,细菌还进化出另外一个防御系统——甲基化酶,利用该酶,细菌把自身重要的同时可能成为"天敌"识别位点或被自己内切酶"误伤"的 DNA 片段保护起来,即把那个位点里的碳原子上的氢原子换成一个更大的甲基(CH_3)。回文序列(如 ctag,cctagg)是最常见的酶切识别位点之一。由此可见,细菌基因组中特定回文字符串缺失或稀少是一个进化产物,作为细菌的防御系统的一个重要一环,不含或少含类似"ctag"的字符串,对其生存繁衍至关重要。反过来说,这也是细菌长期演化过程中在其基因组上留下的痕迹,通过 *K*-mer 分析,我们可以发现这些遗迹。

习 题

1. 简述基于高通量测序短序列基因组拼接和组装过程

2. PE 读序对于基因组拼接有何作用?

3. 请构建下列两条序列的 4 维德布鲁因图(即 $K=4$):

>seq1

ATGGCTCAGTAGGC

>seq2

ATGGCTTTCAGTAGAGAGC

4. 第二代和第三代高通量测序读序有何不同?基因组拼接中如何合理利用这两类数据?

5. 请简述利用字符串(*K*-mer)估计基因组大小的原理

6. 如何利用字符串(*K*-mer)估计基因组的杂合度和倍性?