第三届“量旋杯”公开组竞赛参考样题

本文档内容仅作为学习参考使用，最终的竞赛题目以正式发布为准。

【赛题题目】

设计量子算法解决基于 De Bruijn Graph 的基因组组装中的重复片段问题

【赛题背景】

在现代生物技术的发展过程中，高通量测序技术已经成为获取大规模基因数据的主要手段。然而，由于当前技术的局限性，直接获取完整的基因序列仍然具有挑战性。实际上，我们通常会得到大量的较短的基因片段（也被称为 reads），这些 reads 需要通过某种方式进行拼接，以重建出原始的基因序列。

德布鲁因图（De Bruijn Graph, DBG）算法是一种常用的从头拼接（de novo genome assembly）算法之一。DBG 算法利用图论来处理序列数据，特别是处理基因测序生成的短片段（reads）的问题。给定一系列长度为 k 的字符串（被称为 k-mers），DBG 算法构造一个有向图，其中每个节点表示一个唯一的(k-1)-mer，每条有向边表示一个 k-mer。

其算法组装流程如下：

- *从输入的序列中提取所有的 k-mers。*
- *对于每个 k-mer，创建两个节点，分别表示该 k-mer 的前缀和后缀（即去掉最后一个字符和第一个字符后得到的(k-1)-mer）。*
- *在这两个节点之间添加一条有向边。*
- *对于所有的 k-mer 重复步骤 2 和步骤 3。*

通过这些步骤，我们可以构建出一个 De Bruijn 图，该图捕获了原始序列中的所有 k-mer 的重叠关系。在理想的情况下，我们可以通过寻找这个图中的欧拉路径（即访问每条边一次且仅一次的路径）来重建原始序列。

然而，在实际的组装过程中，由于测序错误和重复基因片段等问题，德布鲁因图通常会更复杂。

测序错误：由于测序技术的误差，reads 可能包含一些错误，例如插入、删除或替换等。这些错误可以引入不存在于真实基因组中的 k-mers，从而在 DBG 中产生错误的边和节点。如图 1 所示，read3 中的胞嘧啶“C”被错误测序为鸟嘌呤“G”，此时 DBG 中产生错误的边和节点。

处理这种情况的一种策略，是利用每个 k-mer 在所有 reads 中出现的次数来定义每条边的权重。直观上，如果一个 k-mer 在许多 reads 中多次出现，那么我们可以相对比较有信心地认为，这个 k-mer 存在于真实的基因组中。反之，如果一个 k-mer 只在很少的 reads 中出现，或者只出现了一次，那么这个 k-mer 很可能是由于测序错误引入的。

根据这个策略，我们可以定义一个优化问题，即寻找一条路径，使得经过的边的权重和的绝对值最大。这条路径将代表我们组装出的基因序列。

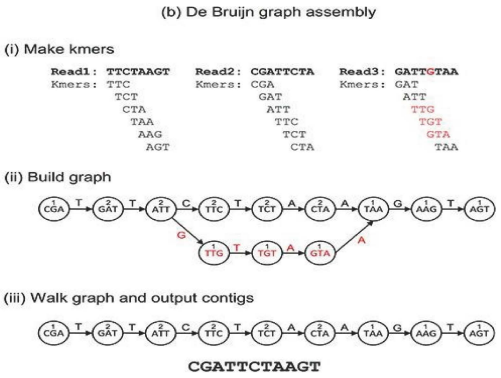

图 1： DBG 算法组装流程（测序错误）

重复片段：许多生物的基因组中都存在大量的重复序列。由于这些重复序列完全相同，我们无法确定 reads 来自哪个重复区域。如图 2 中的序列“ATTC”出现了三次，正常的情况下，我们无法确定一个包含“ATTC”的 read 到底来自于第一次出现的位置、第二次出现的位置还是第三次出现的位置。

为了解决这个问题，一种可能的策略是利用三代测序技术（Third-generation sequencing）。相比于二代测序技术（Second-generation sequencing），三代测序技术可以生成更长的 reads。这意味着，每个 read 能够覆盖更多的基因信息，包括位于重复序列前后的唯一序列。

通过这种方式，我们可以得到足够的上下文信息，从而有可能确定 read 的来源。在组合优化问题中，可以将上下文信息转为奖励项权重，如测序得到 read 为“GATTCC”，则可以确定“ATTC”的 read 来自于第一次出现的位置并分配一个绝对值更大的权重值，最后通过找到权重和绝对值最大的路径代表我们组装出的基因序列。

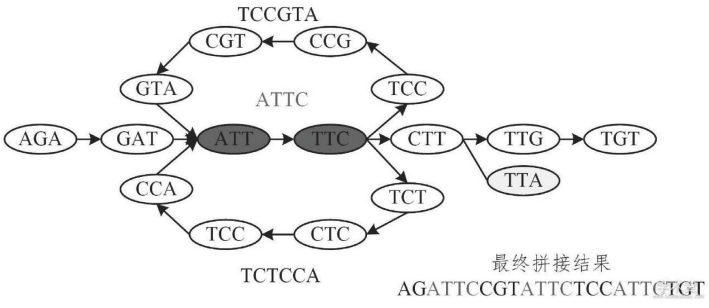

图 2：存在重复片段的 De Bruijn 图及拼接结果

【赛题要求】

请设计一个合适的量子算法将模型（如图 3 所示）进行正确组装。

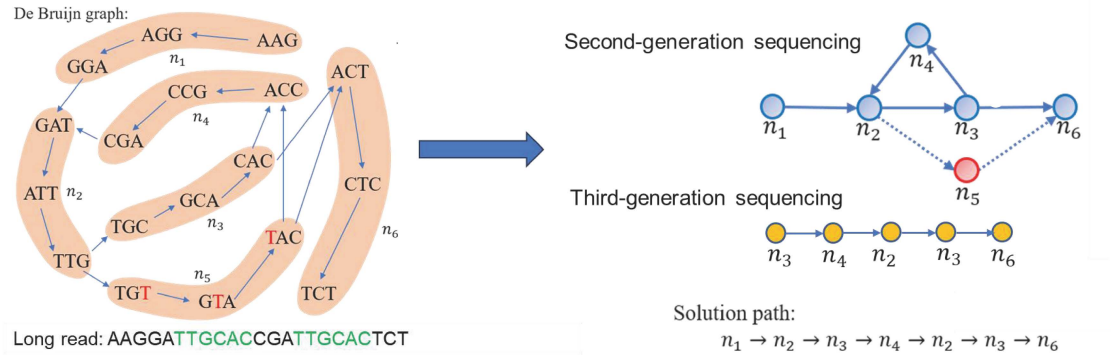

图 3：左图为存在重复片段的德布鲁因图及拼接结果。其中重复片段为“TTGCAC”，测序错误的 read 为“TGTAC”。右图是德布鲁因图的简单示意图及组装路径。

【评分说明】

最终成绩将按照解的正确性和线路的复杂度来进行排名。我们将优先考虑解的正确性，对于得到正确的结果，我们将考虑提交的量子线路的复杂度。尽可能使用少的量子门和多比特量子门的排名靠前。

以上为文档全部内容