arXiv:2602.05193v1 [cs.DS] 5 Feb 2026

序列与泛基因组图之间最长公共子序列相关问题的多项式时间解法

Xingfu Li[1*], **Yongping Wang**[2]1. *College of Big Data Statistics, Guizhou University of Finance and Economics, Guiyang* 550025, *P.R. China;*
2. *Algorithm and Computational Complexity Laboratory of School of Information, Guizhou University of Finance and Economics, Guiyang* 550025, *P. R. China*

摘要 泛基因组能够同时捕捉多个个体间的遗传多样性，相较于单一线性基因组，可为基因组分析提供更全面的参考，从而避免可能产生的等位基因偏倚。广泛采用的一种泛基因组表示形式是节点标记有向图，其中的路径对应于某个物种内可能存在的基因组序列。因此，评估序列与泛基因组图之间的相似性，是泛基因组构造与分析中的一项基本任务。本研究探讨了最长公共子序列（LCS）问题及其涉及序列与泛基因组图的三种变体。我们提出了四种多项式时间归约，将这些最长公共子序列相关问题转化为有向无环图（DAG）中的最长路径问题。这些归约表明，所有四个问题均可在多项式时间内求解，从而证明它们属于复杂性类P。

关键词 最长公共子序列；序列比对；泛基因组图；算法；复杂度

MR（2020）主题分类 05C85；68Q17；68R10；92D20；92-08

1. 引言

泛基因组以系统且全面的方式呈现个体间完整的遗传多样性。与传统线性参考基因组相比，它为基因组研究提供了更全面、无偏的基础，而传统线性参考基因组往往会引入等位基因偏倚。作为物种内所有基因序列的综合库——包括所有个体共有的核心基因，以及仅存在于特定亚群中的附属基因——它已成为计算生物学的重要资源。该框架有助于开展高分辨率比较分析，以揭示遗传变异、推断进化关系并探索不同群体间的功能差异，从而加深我们对生物多样性和适应机制的理解 [7]。表示泛基因组的主流模型是顶点加权有向图 [2]，该模型有助于对不同群体间的基因组变异进行复杂分析。

将序列与泛基因组进行比对已成为计算生物学的核心研究方向，可用于精确衡量查询序列与全面且物种特异的参考泛基因组之间的相似性 [8]。当泛基因组被建模为有向图时，

由贵州省基础研究计划（自然科学）（项目编号：ZK[2022]020）和国家自然科学基金委员会（项目编号：62362007）资助。* 通讯作者电子邮箱：xingfuli@mail.gufe.edu.cn（Xingfu Li）

将序列与泛基因组进行比对，自然引出了序列到泛基因组图比对问题，其正式定义为：在泛基因组图中寻找一条最优路径，使沿该路径重建的序列与给定查询序列的相似性达到最高 [5]。

将一个序列与泛基因组图进行比对，就是识别二者之间的相似性。发现生物序列的相似性是生物信息学中的一个基础问题，并且在癌症诊断 [1] 以及物种共同起源的检测 [13]等诸多应用中受到了广泛关注。在序列比对和分析领域，最长公共子序列（LCS）和最小编辑距离是两种经典的度量方法。尽管二者都用于发现序列之间的相似性或差异性，但作为一种替代性度量，LCS 已被用于衡量序列之间的相似性 [3], ，并在某些场景下展现出独特优势 [12]。其目标是找出两个或多个序列之间的最长公共子序列。为解决序列之间的最长公共子序列问题，人们提出了许多先进算法，例如教材中的动态规划算法 [10], 固定间隙最长公共子序列 [3, 11] 以及带间隙的最长因子 [3]。

本文严格研究了指定序列与泛基因组图之间的最长公共子序列问题，以及与之密切相关的三个计算问题。我们分别将这四个问题归约为有向无环图（DAG）中的最长路径问题，从而证明这四个问题均为多项式时间可解。

本文的组织结构如下。第 2 节介绍基本术语，并给出序列与泛基因组图之间这四个 LCS 相关问题的形式化定义。在第 2 节末尾，我们提出一种高效算法，用于解决有向无环图（DAG）中的最长路径问题，该问题是处理本文所研究的四个 LCS 相关问题的基础子程序。在随后的第 3 节中，我们证明，在次二次时间内无法求出序列与泛基因组图之间的最长公共子序列。随后，我们将这四个 LCS 相关问题分别转换为在特定有向无环图中寻找最长路径的问题，具体内容见第 4 节。我们在第 5 节中通过归约分析解决这些问题的时间复杂度。最后，我们在第6 节总结全文并提出一些未来的研究方向。

2. 预备知识

序列 S的长度（记为 $|S|$）是指其所包含元素的总数。以下标表示的索引指示元素在序列中的位置，并且在本文中，所有索引均从零开始。我们将 $S[i, \ldots, j]$ 定义为 S 中从索引 i 到索引 j（含两端）的子串。例如，考虑序列 $S =$ “$ababc$”；最左侧的“a”位于位置0，而最右侧的“a”出现在位置2。此外，子串$S[1, \ldots, 4]$ 得到“$babc$”。

A pa泛基因组图 $G = (V,\ E,\ \delta)$ 是一个带有顶点标签的有向图，其中 V 表示

顶点集合， E 表示有向边集合，而 δ： $V \to \Sigma^* \setminus \{\epsilon\}$ 是一个标记函数，它为每个顶点分配一个给定字母表 Σ上的非空有限字符串，其中 ϵ表示空字符串。本文将与顶点 $v \in V(G)$ 相关联的序列记为 $\delta(v)$。有向图 G 中的路径 $P = u_0, u_1, u_2, ..., u_k$ 定义为一个顶点序列，其中对于所有 $0 \le i \le k-1$，每一对相邻顶点（u_i, u_{i+1}）均由一条有向边连接。序列 $spell(P) = \delta(u_0)\delta(u_1) \ldots \delta(u_k)$称为路径 P的拼接序列，它通过按路径顺序连接沿路径排列的顶点标签而形成。更确切地说，在泛基因组图中，路径所拼出的序列是依次连接路径上各顶点所关联的序列而得到的。若存在一条从 u 到 v 的有向路径，则称顶点 u 可到达顶点 v，或等价地称 v 可从 u到达，其中路径位于 G中。

2.1 序列与泛基因组图之间的最长公共子序列相关问题

给定一个序列 Q 和一个泛基因组图 G，如果存在一个子序列 S 和一条路径 P，使得在 G 中， S 是 $spell$（P）与 Q之间的公共子序列，则称 S 是 Q 与 G之间的公共子序列。

问题 1（序列与泛基因组图之间的最长公共子序列（LCS-SG）问题）。给定一个序列 Q 和一个泛基因组图 G，求 Q 与 G之间的最长公共子序列。

令 S 表示序列 Q 与泛基因组图 G之间的一个公共子序列。定义 $\text{ind}(S_i, Q)$，其中 $0 \le i \le |S|-1$，为元素 S_i 在 Q中的索引。例如，考虑Q = "$xyabcahde$"和 S = "$acah$"。则，$\text{ind}(S_0, Q) = 2$，$\text{ind}(S_1, Q) = 4$，$\text{ind}(S_2, Q) = 5$，以及 $\text{ind}(S_3, Q) = 6$。如果子序列 S是 Q的子序列，并且存在 G 中的一条路径 P，满足以下条件，则称 S是 Q 与 G之间的（k_1, k_2）-间隔公共子序列：（1） S 是 $spell$（P）与 Q之间的公共子序列；（2）对于每个$1 \le i \le |S|-1$，$0 < ind$（S_i, Q） $- ind$（S_{i-1}, Q） $\le k_1$；以及（3）对于每个 $1 \le i \le |S|-1$， $0 < ind$（S_i, $spell$（P））$- ind$（$S_{i-1}, spell$（P））$| \le k_2$。

问题 2（序列与泛基因组图之间的固定间隙最长公共子序列（FGLCS-SG）问题）。给定一个序列 Q、一个泛基因组图 G 以及两个额外的整数 k_1， $k_2 > 0$。其目标是求出 Q 与G之间的一个最长（k_1， k_2）间隙公共子序列。

设 Q 为一个序列， G =（V、 E、 δ）为一个泛基因组图。若三元组（v、 $[i_h, \ldots, i'_h]$、$[j_h, \ldots, j'_h]$）满足 δ（v）$[i_h, \ldots, i'_h]$ =$Q[j_h, \ldots, j'_h]$，其中 $v \in V$（G），则称其为 Q与 G之间的一个精确匹配（或简称匹配或种子）。若匹配（v、 $[i_h, \ldots, i'_h]$、 $[j_h, \ldots, j'_h]$）满足 δ（v）$_{i_h-1}$ $\ne Q_{j_h-1}$或 $i_h = 0$或 $j_h = 0$，则称其为左极大。若匹配（v、 $[i_h, \ldots, i'_h]$、 $[j_h, \ldots, j'_h]$）满足 δ（v）$_{i'_h+1} \ne Q_{j'_h+1}$或 $i'_h = |\delta(v)| - 1$或 $j'_h = |Q| - 1$，则称其为右极大。如果一个匹配既是左极大又是右极大，则称其为极大精确匹配（MEM）。

对于给定的匹配 m =(v, $[i_h, \ldots, i'_h]$, $[j_h, \ldots, j'_h]$)，我们将其组成部分定义如下： m_0表示顶点 v， m_1 表示索引序列 $[i_h, \ldots, i'_h]$，而 m_2 对应序列 $[j_h, \ldots, j'_h]$。此外，符号 $|m_1|$ 和 $|m_2|$ 用于表示

m_1 和 m_2中包含的索引数量，分别表示各索引列表的长度或大小。请注意， m_1 的长度等于 m_2的长度，即 $|m_1| = |m_2|$，因为 m是精确匹配。

两个种子（v_1，$[i_1, \ldots, i'_1]$，$[j_1, \ldots, j'_1]$）和（v_2，$[i_2, \ldots, i'_2]$，$[j_2, \ldots, j'_2]$）表示查询序列 Q 与泛基因组图 G之间的局部精确匹配。当满足 $j'_1 < j_2$且以下条件之一成立时，称它们为严格有序或处于严格顺序：如果两个种子映射到图 G中的同一个节点，即 $v_1 = v_2$，那么 $i'_1 < i_2$；否则，如果 $v_1 \neq v_2$，节点v_1 必须能够通过泛基因组图G的有向结构中的有效路径到达 v_2 ，从而确保保持一致的基因组顺序。本文将这两个严格有序的种子记作（v_1，$[i_1, \ldots, i'_1]$，$[j_1, \ldots, j'_1]$）$\prec$（v_2，$[i_2, \ldots, i'_2]$，$[j_2, \ldots, j'_2]$）。

令 $\mathcal{M}$ 表示序列 Q 与泛基因组图 G之间 MEM 的一个子集。 $\mathcal{M}$ 的长度定义为 $len(\mathcal{M}) = \sum_{m \in \mathcal{M}} |m_1| = \sum_{m \in \mathcal{M}} |m_2|$。如果可以将 $\mathcal{M}$ 中的种子线性排列，使得每一对相邻种子都严格有序，那么 $\mathcal{M}$就称为严格有序子集。

问题 3（MEM 链（MEMC）问题）。给定 MEM 的一个子集 $\mathcal{M}$，以及一个泛基因组图 $G = (V, E, \delta)$，从 $\mathcal{M}$ 中找出一个严格有序子集 $\mathcal{M}'$，使得 $len(\mathcal{M}')$ 最大。

如果我们希望在泛基因组图中找到一条路径，使其尽可能多地经过匹配，而不考虑与每个顶点关联的序列长度，那么我们就会提出最大种子问题。

问题 4（最大种子问题（MSP））。给定一个序列 Q；一个泛基因组图 $G = (V, E, \delta)$；以及 Q与 G之间的一组种子 $\mathcal{S}$。MSP 要求在 G 中找到一条路径，使其按严格顺序经过 $\mathcal{S}$ 中的最多种子。

2.2 有向无环图上的最长路径

有向无环图（DAG）是一种每条边都有方向且不存在环的图。这意味着，从任意顶点出发沿有向边遍历，都无法回到起点。有向无环图具有许多有趣的性质，广泛用于表示现实世界中的结构，例如任务依赖关系和版本控制系统。

在有向无环图中，边加权最长路径问题是指找出从任意源顶点到任意汇顶点的最长路径，其中路径长度定义为其边权之和。尽管在有向循环图中可以高效地计算最短路径，例如使用 Dijkstra算法 [6]，但在这类图中，最长路径问题是 NP 难的 [9]。幸运的是，有向无环图中不存在环，因此可以高效地求解最长路径问题。

在有向无环图中求解最长路径问题，关键在于结合两种强大的技术：拓扑排序，它按照边的方向对顶点进行排序；以及动态规划，它通过复用先前的结果逐步构建解。

拓扑排序是指有向无环图中顶点的一种线性排序，使得对于每条有向边（u，v），顶点 u 在序列中都先于顶点 v 。这种排序仅适用于有向无环图——

因为环会使有效排序无法成立，而且值得注意的是，拓扑序并不唯一——可能存在多个有效序列。给定有向无环图上的拓扑排序 G 可以通过以下过程高效地在 $O(|V(G)|+|E(G)|)$时间内计算得出 [6]。首先，对 G 执行深度优先搜索，以确定 G中每个顶点的完成访问时间。然后，每个顶点完成访问后，将其插入链表的开头。所得链表即为顶点的拓扑排序。该方法的正确性及其线性时间复杂度已在 [6]中得到证明。

让我们探讨如何有效地运用动态规划来确定最长路径。核心思想是按拓扑序处理顶点，同时动态跟踪以每个顶点为终点的最长路径长度。令 Q 为给定有向无环图 G 的拓扑排序顶点列表。令 $dist^e(v)$ 为以顶点 v 为终点的边加权最长路径长度。如果在图 G 中存在一条从 u 到 v 的有向边，那么顶点 u 是另一个顶点 v 的入邻居。令 $N^-(v)$ 为图 G 中顶点 v 的入邻居集合。由于 G 中的每条边都遵循 Q 的顺序，因此迭代关系 2.1 对每个索引 i 都成立：
$0 \le i < |Q| - 1$ 属于 Q:

$$dist^e(Q_i) = \begin{cases} 0 & N^-(Q_i) = \emptyset \\ \max_{u \in N^-(Q_i)} \left\{ dist^e(u) + weight\big((u, Q_i)\big) \right\} & N^-(Q_i) \neq \emptyset \end{cases}, \tag{2.1}$$

其中，对于 $1 \le i < |Q| - 1$， $u \in N^-(Q_i)$ 的索引在 Q 中必须小于 i。

用于求解有向无环图上的边加权最长路径问题的完整算法如算法1所示。由于该算法对每个顶点和边的处理次数均为常数，因此对于任意给定的有向无环图 $O\big(|V(G)| + |E(G)|\big)$，其时间复杂度为 G。

算法 1： 有向无环图上的边加权最长路径

输入： 一个有向无环图 $G = (V, E)$，其中每条边 $(u, v) \in E$ 都具有非负权重 $weight\big((u, v)\big)$.

输出： G中的一条边加权最长路径。

```
对 G进行拓扑排序;
分别初始化数组 dist 和 π，用于存储最长路径长度以及以每个顶点为终点的最长路径中的父
  每个顶点处结束的最长路径中的顶点;
为每个顶点 v 初始化设置 dist[v] = 0 和 π[v] = ∅;
对拓扑序中的每个顶点 u 执行
    对每个 u 的出邻居 v;          /* 若顶点 v 是 u的出邻居,
     则存在一条从 u 指向 v的有向边。          */
    do如果 dist[u] + weight((u, v)) > dist[v]，则
        更新 dist[v] = dist[u] + weight((u, v));
        π[v] ← u;
    end
  end
结束
返回 dist 数组和列表 π 中的最大值。
```

Give在每个顶点都被赋予非负权重的有向无环图中，顶点加权最长

路径问题旨在找出一条路径，使其沿途顶点权重之和最大。令 $dist^v(u)$ 表示以顶点 u 结尾的所有路径中的最大顶点权重总和，即以 u 结尾的最长顶点加权路径长度。

顶点加权最长路径问题中，以每个顶点为终点的最长路径长度呈现出迭代关系，如公式 2.2 所示，这与公式2.1中所示的边加权情形类似。

$$dist^v(Q_i) = \begin{cases} weight(Q_i) & N^-(Q_i) = \emptyset \\ \max_{u \in N^-(Q_i)} \left\{ dist^v(u) + weight(Q_i) \right\} & N^-(Q_i) \neq \emptyset \end{cases} \tag{2.2}$$

算法2给出了在有向无环图中求解顶点加权最长路径问题的完整流程。与算法 1类似，对于给定的有向无环图 $O\big(|V(G)|+|E(G)|\big)$，其时间复杂度为 G，因此在实践中效率很高。

算法 2: 有向无环图上的顶点加权最长路径

输入：一个有向无环图 $G = (V, E)$，其中每个顶点 $v \in V$的权重均为非负权重 $weight(v)$。输出： G中的一条顶点加权最长路径。1 对图进行拓扑排序；2 初始化数组 $dist$和 π，分别用于存储以每个顶点为终点的最长路径长度及该最长路径中的父顶点；3 对每个顶点 v，初始设置 $dist[v] = weight(v)$ 和 $\pi[v] = \emptyset$；4 按拓扑序遍历每个顶点 u；5 遍历 u 的每个出邻居 v；/* 如果存在一条从 u 到 v的有向边，则顶点 v 是 u 的出邻居。*/ 6 执行 7 如果 $dist[u] + weight(v) > dist[v]$，则 8 更新 $dist[v] = dist[u] + weight(v)$；
9 $\pi[v] \leftarrow u$；10 结束 11 结束 12 结束 13 返回 $dist$ 数组中的最大值及列表 π。

3. 序列与泛基因组图之间最长公共子序列问题的下界

两个序列之间的最长公共子序列（LCS）问题是计算机科学和生物信息学中的经典难题。已知在 [4], 强指数时间假设（SETH）成立的条件下，该问题不存在次二次时间算法；这意味着在实践中，无法找到显著更快且能精确求解该问题的算法。这种计算难度对许多需要高效序列比对的应用具有重要影响，例如基因组分析和文本处理。

给定字母表 Σ上的两个序列 $Q_1, Q_2 \in \Sigma^*$，其中 Σ^* 表示由 Σ构成的所有可能字符串的集合，我们构造一个仅包含单个顶点 v 的泛基因组图 G，并为其指定

标签 $\delta(v) = Q_2$，从而将其中一个序列直接编码到图结构中。将 Q_1 与这个简化的图表示进行比对后，我们便将传统的最长公共子序列问题转化为一种基于图的变体，从而得到一个LCS-SG问题实例。该问题将经典表述扩展到了更复杂的基因组表示。

引理1。 当且仅当存在一个长度至少为 ℓ 的 Q_1 与 Q_2 的最长公共子序列，且存在一个长度至少为 ℓ的 Q_1 与 G 的ˆ最长公共子序列。

ˆ证明 由于图 G 只包含一个顶点，且与该顶点关联的序列是两个序列 Q_2之一，因此 Q_1 与 Q_2 的公共子序列等价于 Q_1 与 G的公共子序列，反之亦然。由此，引理得证。

定理1。 在*SETH*成立的条件下，*LCS-SG*问题无法在次二次时间内求解。

ˆ证明 设 Q_1 和 Q_2 是字母表上的两个序列。构造 G 首先创建一个顶点，然后将 Q_2 指定为与该顶点关联的序列。 G中没有任何边。因此，该构造总共只需线性时间。此外，根据引理 1，该定理得以严格证明。

推论1. 在 *SETH* 成立的条件下，*FGLCS-SG* 问题无法在次二次时间内求解。

证明 对于 LCS-SG 问题的实例，我们进一步引入两个参数 $k_1 = +\infty$ 和$k_2 = +\infty$，从而将其有效地转化为 FGLCS-SG 问题。换言之，LCS-SG 问题是 FGLCS-SG 问题的一个子问题，区别在于前者没有间隙约束。根据定理 1，FGLCS-SG 问题无法在次二次时间内求解，这为其计算复杂性确立了一个基本下界。

4. 问题变换

首先，我们研究 LCS-SG 问题。我们构造如下新的有向图 H：

顶点集定义为 $V(H) = \{v_{(i,u,f)} \mid 0 \le i \le |Q|-1, u \in V(G)$，且 $Q_i = \delta(u)f$，其中 $0 \le f \le |\delta(u)|-1\}$。

边集为 $E(H) = \{(v_{(i,u,f)}, v_{(i',u',f')}) \mid 0 \le i < i' \le |Q|-1, u, u' \in V(G)\ 0 \le f \le |\delta(u)|-1, 0 \le f' \le |\delta(u')|-1$，且满足以下任一条件： $u \ne u'$ 存在一条从 u 到 u' 的路径，该路径位于 G，，中；或者 $u = u'$ 且 $f < f'.\}$

引理 2. H 是一个有向无环图。

证明 假设存在一个环 $C = v_{(i_0,u_0,f_0)}\ v_{(i_1,u_1,f_1)}\ v_{(i_2,u_2,f_2)} \cdots, v_{(i_k,u_k,f_k)}\ v_{(i_0,u_0,f_0)}$。根据 H的构造，我们得到 $i_0 < i_1 < i_2 < \cdots < i_k < i_0$，这意味着 $i_0 < i_0$，从而导致矛盾。因此， H 必定是一个有向无环图。

引理 3. H 中的一条有向路径表示 Q 和 G 之间的一个公共子序列 .

证明 设 $P = u_{(i_0,u_0,f_0)}\ u_{(i_1,u_1,f_1)}\ \cdots, u_{(i_k,u_k,f_k)}$ 是 H 中的一条有向路径。根据 H 的构造，对于每个顶点 $u_{(i_j,u_j,f_j)} \in V(P)$：$0 \le j \le k$，都有 $Q_{i_j} = \delta(u_j)f_j$。此外，对于每一对 $\big(u_{(i_j,u_j,f_j)}\ u_{(i_{j+1},u_{j+1},f_{j+1})}\big)$：$0 \le j < k$，都有 $i_j < i_{j+1}$；如果$u_j \neq u_{j+1}$，那么 u_j 可以到达 u_{j+1}；否则，$f_j < f_{j+1}$。因此，这表明序列$Q_{i_0}, Q_{i_1}, \ldots, Q_{i_k}$是 Q 和 G 之间的一个公共子序列。

引理 4. *Q 与 G 之间的每个公共子序列都表示 H中的一条有向路径。*

证明 设 S 是 Q 与 G之间的一个公共子序列。对于 $0 \le i \leq |S| - 1$，设 $\rho(i) = ind(S_i,\ Q)$。由于对于序列 S中的 $0 \le i \le |S| - 2$，S_i 必须位于 S_{i+1} 之前，因此它们在 Q 中的出现顺序与在 S中的相同，即 $\rho(i) < \rho(i+1)$。

设 P 为 G 中的路径，使得 S 是 $spell(P)$与 Q之间的一个公共子序列。设$X = spell(P)$。对于 $0 \le i \le |S| - 1$，设 $\phi(i) = ind(S_i,\ X)$。设 $u_{\phi(i)} \in V(G)$：$0 \le i \le |S| - 1$为 $X_{\phi(i)}$ 的来源顶点。对于 $0 \le i \le |S| - 1$，设 $f_{\phi(i)} = ind\big(X_{\phi(i)},\ \delta(u_{\phi(i)})\big)$。

如果$u_{\phi(i)} = u_{\phi(i+1)}$，则 $f_{\phi(i)} < f_{\phi(i+1)}$，因为 $X_{\phi(i)}$ 在序列 X中位于 $X_{\phi(i+1)}$ 之前。因此，在图 H中，对于每个 $0 \le i \le |S| - 2$，存在一条从$v_{(\rho(i),u_{\phi(i)},f_{\phi(i)})}$ 到顶点$v_{(\rho(i+1),u_{\phi(i+1)},f_{\phi(i+1)})}$的有向边。

如果 $u_{\phi(i)} \neq u_{\phi(i+1)}$，则$u_{\phi(i)}$ 可以通过路径 P到达$u_{\phi(i+1)}$。因此，在图H中，对于每个 $0 \le i \le |S| - 2$，存在一条从$v_{(\rho(i),u_{\phi(i)},f_{\phi(i)})}$ 到顶点$v_{(\rho(i+1),u_{\phi(i+1)},f_{\phi(i+1)})}$ 的有向边。

因此，$v_{(\rho(0),u_{\phi(0)},f_{\phi(0)})}$、$v_{(\rho(1),u_{\phi(1)},f_{\phi(1)})}$、…、$v_{(\rho(|S|-1),u_{\phi(|S|-1)},f_{\phi(|S|-1)})}$ 是 H中的一条有向路径。

定理 2. *H 中的一条最长路径表示 Q 和 G之间的一个最长公共子序列。*

证明 根据引理 4，Q和 G 之间的每个公共子序列都由 H中的一条有向路径表示 H。另一方面，根据引理 Q 和 G，3，H 中的每条有向路径都表示 Q 和 G 之间的一个公共子序列。因此，H 中的一条最长路径必然对应 Q 和 G之间的一个最长公共子序列。

现在考虑 FGLCS-SG 问题。我们按如下方式构造一个新的有向图 H_{gap}：

顶点集定义为 $V(H_{gap}) = \{\ v_{(i,u,f)}$：$0 \le i \le |Q| - 1,\ u \in V(G)$，使得$Q_i = \delta(u)f$ 对于某个 $0 \le f \le |\delta(u)| - 1$ $\}$成立。

边集定义为 $E(H_{gap}) = \{\ (v_{(i,u,f)},\ v_{(i',u',f')})$：$0 < i' - i \le k_1,\ u,\ u' \in V(G)$，$0 \le f < |\delta(u)|, 0 \le f' < |\delta(u')|$，并且若 $u \neq u'$，则在 G 中存在一条路径 P，从 u 到 u'，使得$0 < ind\big(\delta(u')f',\ spell(P)\big) - ind\big(\delta(u)f,\ spell(P)\big) \le k_2$；否则 $0 < f' - f \le k_2$ $\}$。

引理 5. *H_{gap} 是一个有向无环图。*

证明 比较 H_{gap} 和 H，以探究它们的结构特征。两者具有相同的顶点集，且 H_{gap} 的边集包含于 E（H）中，因此 H_{gap} 是 H的子图。所以 H_{gap} 也是一个有向无环图。

引理 6. H_{gap} 中的一条有向路径表示 Q 和 G之间的（k_1, k_2）-公共子序列 .

证明 根据引理 5， H_{gap} 是有向无环图，也是 H的子图。由引理 3可知， H_{gap} 中的每条有向路径都表示 Q 和 G之间的一个公共子序列。此外，根据 H_{gap}的构造， H_{gap} 中的一条有向路径表示 k_1， k_2)-公共子序列，介于Q和 G之间。

引理 7. Q 与 G 之间的每个（k_1， k_2）-公共子序列都表示H_{gap}中的一条有向路径。

证明 设 S 是 Q 与 G之间的一个（k_1， k_2）-公共子序列。那么，在 G 中存在一条路径 P，使得 S 是 $spell$（P）与 Q之间的一个公共子序列。设 $X = spell$（P）。对于$0 \le i \le |S| - 1$，设 ϕ（i） $= ind$（S_i， X）。此外，对于 $0 \le i \le |S| - 1$，设 ρ（i） $= ind$（S_i， Q）。设 $u_{\phi(i)}$ 是 G 中的顶点，元素 $X_{\phi(i)}$ 来源于该顶点，并且 $f_{\phi(i)} = ind\big(X_{\phi(i)}$， δ $(u_{\phi(i)})\big)$。

对于任意两个相邻元素 S_i 和 S_{i+1}，其中 $0 \le i \le |S| - 2$，由于 S 是 Q 和 G之间的一个（k_1， k_2）公共子序列，因此我们有 $\rho(i+1) - \rho(i) \le k_1$，且 $\phi(i+1) - \phi(i) \le k_2$。现在我们将证明，对于$0 \le i < |S| - 1$，存在一条从 $v_{(\rho(i),u_{\phi(i)},f_{\phi(i)})}$ 到 $v_{(\rho(i+1),u_{\phi(i+1)},f_{\phi(i+1)})}$ 的边。

如果 $u_{\phi(i)} = u_{\phi(i+1)}$，那么 $X_{\phi(i)}$ 和 $X_{\phi(i+1)}$ 位于同一个顶点上。于是 $f_{\phi(i)} = \phi(i)$ 且 $f_{\phi(i+1)} = \phi(i+1)$。由于 $\phi(i+1) - \phi(i) \le k_2$，因此我们有 $f_{\phi(i+1)} - f_{\phi(i)} \le k_2$。如果 $u_{\phi(i)} \neq u_{\phi(i+1)}$，由于 $u_{\phi(i)}$ 和 $u_{\phi(i+1)}$ 都位于路径 P上， $u_{\phi(i)}$ 可以在 G中到达 $u_{\phi(i+1)}$。

因此，对于 $0 \le i < |S| - 1$，存在一条从 $v_{(\rho(i),u_{\phi(i)},f_{\phi(i)})}$ 到 $v_{(\rho(i+1),u_{\phi(i+1)},f_{\phi(i+1)})}$ 的边。从而 S表示 H_{gap} 的一条有向路径。

定理 3. H_{gap} 中的一条最长路径表示一条最长的(k_1, k_2)-公共子序列，介于Q 与 G之间。

证明 根据引理 7，每一条（k_1， k_2）-公共子序列，介于 Q 与 G 之间，表示一条有向路径，位于 H_{gap}。另一方面，每条有向路径，位于 H_{gap} ，表示一条（k_1， k_2）-公共子序列，介于 Q 与 G 之间，根据引理 6。因此，一条最长路径，位于 H_{gap} ，必定导出一条最长（k_1， k_2）-公共子序列，介于 Q 与 G。

现在让我们考虑 MEMC 问题。回想一下，MEMC 的实例是 MEM 的一个子集 $\mathcal{M}$ 以及一个附加的泛基因组图 G。其目标是从 $\mathcal{M}$ 中找出一个严格有序子集 $\mathcal{M}'$，使得 $len(\mathcal{M}')$ 最大。我们按如下方式构造一个顶点加权有向图 H_{MEM}： $V(H_{MEM}) = \{v^m$： $m \in \mathcal{M}\}$； $E(H_{MEM}) = \big\{(v^m, v^{m'})$： $m \prec m'$ 与 $m, m' \in \mathcal{M}\big\}$。每个顶点 $v^m \in V(H_{MEM})$ 的权重定义为 $W(v^m) = |m_1|$。

引理 8。 H_{MEM} 是一个有向无环图。

证明 反设存在一个环，即 $C = v^{m^1}$、 v^{m^2}、 …、 v^{m^k}、 v^{m^1}，位于 H_{MEM} 中。那么我们有 $m^1 \prec m^2 \prec \ldots \prec m^k \prec m^1$。

考虑 m_2^i 对于 $1 \le i \le k$，其中符号 m_0， m_1 和 m_2 均在第 2.1节中定义。设 $m_2^1 = [j_1, \ldots, j_1']$， $m_2^2 = [j_2, \ldots, j_2']$， ……， $m_2^k = [j_k, \ldots, j_k']$。由于 $m^1 \prec m^2 \prec$

$\ldots \prec m^k \prec m^1$，我们有 $j'_1 < j_2, j'_2 < j_3, \ldots\ldots, j'_{k-1} < j_k, j'_k < j_1$。显然，$j_2 \leq j'_2, \ldots\ldots, j_k \leq j'_k$。因此我们有 $j'_1 < j_2 \leq j'_2 < j_3 \leq \ldots \leq j'_{k-1} < j_k \leq j'_k < j_1$。这与 $j_1 \leq j'_1$ 这一事实矛盾。

因此，H_{MEM} 是一个有向无环图。

引理9. H_{MEM} 中的每条路径都表示 $\mathcal{M}$的一个严格有序子集。

证明 设 $v^{m^1}, v^{m^2}, \ldots, v^{m^k}$ 是 H_{MEM}中的一条路径。根据 H_{MEM}的构造，我们有 $m^1 \prec m^2 \prec \ldots \prec m^k$。因此，$\{ m^i : 1 \leq i \leq k \}$ 是 $\mathcal{M}$的一个严格有序子集。

引理10. $\mathcal{M}$ 的每个严格有序子集都表示 H_{MEM}中的一条路径。

证明 设 $\{ m^i : 1 \leq i \leq k \}$ 是 $\mathcal{M}$的一个严格有序子集。不失一般性，设 $m^1 \prec m^2 \prec \ldots \prec m^k$。那么，对于每个 $1 \leq i \leq k-1$，v^{m^i} 到 $v^{m^{i+1}}$ 在H_{MEM}中都有一条边。因此，序列$v^{m^1}, v^{m^2}, \ldots, v^{m^k}$ 是 H_{MEM}中的一条路径。

定理4。 H_{MEM} 中的顶点加权最长路径表示一个严格有序子集 $\mathcal{M}'$，其$\mathcal{M}$ 的 $len(\mathcal{M}')$达到最大值。

证明 根据引理9 和10，H_{MEM} 中的路径与 $\mathcal{M}$ 的严格有序子集一一对应。此外，回顾 H_{MEM} 的构造，每个顶点$v^m \in V(H_{MEM})$的权重等于 m_1的长度。因此，H_{MEM} 中的顶点加权最长路径表示一个严格有序子集 $\mathcal{M}'$，其 $\mathcal{M}$ 的 $len(\mathcal{M}')$达到最大值。

现在我们来考虑 MSP 问题。MSP 问题的实例包括查询序列 Q、泛基因组图 G，以及它们之间的一组种子 $\mathcal{S}$。我们按如下方式构造一个顶点加权有向无环图H_{MSP}：$V(H_{MSP}) = \{ v^s : s \in \mathcal{S} \}$；$E(H_{MSP}) = \{ (v^s, v^{s'}) : s \prec s' \text{ 和 } s, s' \in \mathcal{S} \}$。每个顶点 $v^s \in V(H_{MSP})$ 的权重定义为 $W(v^s) = 1$。

定理5。 H_{MSP} 中的顶点加权最长路径表示 $\mathcal{S}$中种子数量最多的严格有序子集。

证明 其顶点集和边集 H_{MSP} 的构造方式与H_{MEM}相同。MSP 要求找到种子数量最多的子集，这等价于在 H_{MSP}中寻找顶点加权最长路径。因此，正确性可直接由定理4推出。

现在，我们已将四个问题——LCS-SG、FGLCS-SG、MEMC 和 MSP——归约为有向无环图中的最长路径问题。下一节将对每个问题的时间复杂度进行详细而全面的分析。

5. 复杂度分析

根据定理2和3，LCS-SG 和 FGLCS-SG 的最优解分别对应于 H 和 H_{gap}中的边加权最长路径。类似地，根据定理4 和5，MEMC 和 MSP 的最优解分别是 H_{MEM} 和 H_{MSP}中的顶点加权最长路径。第2.2 节说明，边加权和顶点加权最长

有向无环图中的路径问题可以在线性时间内求解。下面，我们分别对 H、 H_{gap}、 H_{MEM} 和 H_{MSP} 的构造进行详细的时间复杂度分析。

给定一个序列和一个泛基因组图 G，构造 $V(H)$ 需要确定每个 Q_i 是否与某个 $u \in V(G)$ 以及某个 $0 \le f \le |\delta(u)| - 1$ 匹配 $\delta(u)_f$，其中 $0 \le i \le |Q| - 1$。因此，构建 $V(H)$ 的时间复杂度为 $O(|Q|N)$，其中 N 表示 G 中所有顶点上的序列总长度，具体来说是 $N = \sum_{v \in V(G)} |\delta(v)|$。

构造 $E(H)$ 需要确定 $V(H)$ 中任意一对顶点之间是否存在边。回顾一下，$|V(H)| = O(|Q|N)$。设 $v_{(i,u,f)}$ 和 $v_{(i',u',f')}$ 是 $V(H)$ 中的两个顶点。当 $u = u'$ 时，只需比较 i 和 i'，以及 f 和 f'，这可以在 $O(1)$ 时间内完成。当 $u \neq u'$ 时，不仅需要比较 i 和 i'，还需要检查 u 和 u' 之间的可达性。因此，在构造 $E(H)$ 之前，我们必须确定 $V(G)$ 中每对顶点之间的可达性。众所周知，使用求全源最短路径的 Floyd-Warshall 算法，可以在 $O(n^3)$ 时间内完成，其中 n 表示图中的顶点数。因此，构造 $E(H)$ 的总体时间复杂度为 $O(n^3 + |Q|^2N^2)$，其中 $O(|Q|^2N^2)$ 是 $V(H)$ 中顶点对的总数。

给定序列 Q、泛基因组图 G 以及另外两个整数 k_1, k_2，顶点集 $V(H_{gap})$ 的构造方式与 $V(H)$ 相同。我们现在分析构造 $E(H_{gap})$ 所涉及的时间复杂度。考虑 $V(H_{gap})$ 中的两个顶点 $v_{(i,u,f)}$ 与 $v_{(i',u',f')}$。当$u = u'$时，只需验证 $0 < i' - i \le k_1$ 与 $0 < f' - f \le k_2$ 是否成立，该操作仅需 $O(1)$ 时间。

若 $u \neq u'$，则不仅需要验证 $0 < i' - i \le k_1$，还需要确保$0 < \mathrm{ind}(\delta(u')_{f'}, \mathrm{spell}(P)) - \mathrm{ind}(\delta(u)_f, \mathrm{spell}(P)) \le k_2$。为此，我们将给定的泛基因组图转换为单字符图：把每个顶点 $v \in V(G)$ 拆分为 $|\delta(v)|$ 个独立顶点，同时保留各自的来源。所得图记为 G^s。G^s 的阶恰好等于 $N = \sum_{v \in V(G)} |\delta(v)|$，其规模为 $|E(G)| + N - n$。为验证是否存在一条路径 P，从某个顶点 $u \in V(G)$ 到某个顶点 $u' \in V(G)$，使得对某个 $0 \le f < |\delta(u)|$ 以及某个 $0 \le f' < |\delta(u')|$ 有 $0 < \mathrm{ind}(\delta(u')_{f'}, \mathrm{spell}(P)) - \mathrm{ind}(\delta(u)_f, \mathrm{spell}(P)) \le k_2$，只需检查更新后的图 G^s 中标记为 $\delta(u)_f$ 的顶点与标记为 $\delta(u')_{f'}$ 的顶点之间的距离是否不超过 k_2。这对应于图 G^s中的全源最短路径问题，可用 Floyd-Warshall 算法求解，耗时 $O(N^3)$ [6]。因此，构造 $E(H_{\mathrm{gap}})$ 需要 $O(N^3 + |Q|^2N^2)$时间。

给定一组 MEM（$\mathcal{M}$）和一个泛基因组图（G），顶点集 $V(H_{MEM}) = \{v^m : m \in \mathcal{M}\}$ 可在 $O(|\mathcal{M}|)$ 时间内构造出来。现在我们来分析构造 $E(H_{MEM})$ 的时间复杂度。考虑两个不同的 MEM，$m \in \mathcal{M}$ 和 $m' \in \mathcal{M}$。判断是否存在从 v^m 到 $v^{m'}$ 的边，相当于检查 $m \prec m'$。回顾第 2.1 节中“$\prec$”的定义。当 $m_0 = m'_0$ 时，只需比较列表 m_1 的最后一个元素和列表 m'_1 的第一个元素，这可以在 $O(1)$ 时间内完成。当 $m_0 \neq m'_0$ 时，只需在图 G 中检查从 m_0 到 m'_0 的可达性，从而将其转化为标准的图可达性问题。Floyd-Warshall 算法需要 $O(n^3)$ 时间来

求解 G [6] 中所有顶点对之间的可达性问题，其中 $n = |V(G)|$。因此，构造 $E(H_{MEM})$ 需要 $O(n^3 + |\mathcal{M}|^2)$ 时间。

最后， H_{MSP} 的构造过程与 H_{MEM}相同，因此时间复杂度为 $O(n^3 + |\mathcal{S}|^2)$，其中 n 表示给定泛基因组图中的顶点数量， $\mathcal{S}$ 表示 MSP 实例中的种子集。

6. 结论与讨论

本文研究了序列与泛基因组图之间的最长公共子序列问题及其另外三种变体，全面分析了这些问题的计算特性及其意义。我们证明，这四个问题均可通过归约为有向无环图中的最长路径问题，在多项式时间内求解；这一结果借助成熟的图论方法，彰显了这些问题在计算上的可处理性。

然而，对于大规模实例而言，立方时间复杂度在实践中极不实用，因为它会导致过长的计算时间和大量的资源消耗，从而给实际应用带来重大挑战。随着各领域的生物序列数据不断增长，如此高的时间复杂度所带来的弊端日益明显，妨碍了问题得到及时、高效的解决。为克服这一局限，未来研究应优先设计和开发时间复杂度低于立方，甚至低于二次的高效算法，从而为复杂计算挑战提供可扩展且实用的解决方案。亚立方时间复杂度指算法的运行时间随输入规模 n增长的速度慢于立方函数 $O(n^3)$。

参考文献

[1] Alexander M. Aravanis、Mark A. Lee 和 Richard D. Klausner。利用循环肿瘤DNA进行下一代测序以实现早期癌症检测。 *Cell*，168:571–574，2017。[2] Jasmijn A Baaijens、Paola Bonizzoni、Christina Boucher、Gianluca Della Vedova、Yuri Pirola、Raffaella Rizzi 和 Jouni Sir´en。计算图泛基因组学：数据结构及其应用教程。 *Natural computing*，21(1):81–108，2022。[3] Aranya Banerjee、Daniel Gibney 和 Sharma V Thankachan。带间隙的最长公共子串及相关问题。载于 *32nd Annual European Symposium on Algorithms (ESA 2024)*，第16–1页。Schloss Dagstuhl–Leibniz-Zentrum f¨ur Informatik，2024。[4] Karl Bringmann 和 Marvin K¨unnemann。字符串问题和动态时间规整的二次条件下界。载于 *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*，第79–97页。IEEE，2015。[5] Ghanshyam Chandra 和 Chirag Jain。适用于无环泛基因组图的间隙敏感共线链式算法。 *Journal of Computational Biology*，30(11):1182–1197，2023。[6] Thomas H Cormen、Charles E Leiserson、Ronald L Rivest 和 Clifford Stein。 *Introduction to algorithms*。MIT出版社，2022。[7] Jana Ebler、Peter Ebert、Wayne E Clarke、Tobias Rausch、Peter A Audano、Torsten Houwaart、Yafei Mao、Jan O Korbel、Evan E Eichler、Michael C Zody 等。基于泛基因组的基因组推断能够在广泛的变异类别中实现高效且准确的基因分型。 *Nature genetics*，54(4):518–525，2022。[8] Jordan M Eizenga、Adam M Novak、Jonas A Sibbesen、Simon Heumos、Ali Ghaffaari、Glenn Hickey、Xian Chang、Josiah D Seaman、Robin Rounthwaite、Jana Ebler 等。泛基因组图。 *Annual review ofgenomics and human genetics*，21(1):139–162，2020。

[9] Michael R Garey 和 David S Johnson。 *Computers and intractability*，第29卷。wh freeman New York，2002。[10] Daniel S Hirschberg。最长公共子序列问题的算法。 *Journal of the ACM (JACM)*，24 4 664( ): –675，1977。[11] Costas S Iliopoulos 和 M Sohel Rahman。计算最长公共子序列问题各种变体的算法。 *Theoretical Computer Science*，395(2-3):255–267，2008。[12] Kristoffer Sahlin。利用随机种子片段进行有效的序列相似性检测。 *Genome research*，31(11):2080–2094，2021。[13] Marketa Zvelebil 和 Jeremy O Baum。 *Understanding bioinformatics*。Garland Science，2007。