Niu 等人. BMC 生物信息学 (2022 年) 23:219
https://doi.org/10.1186/s12859-022-04712-z

BMC 生物信息学

研究 开放获取

ACO：基于自适应编码顺序的无损质量分数压缩

Yi Niu[1,2*], Mingming Ma[1], Fu Li[1], Xianming Liu[2] and Guangming Shi[1]

*通信：niuyi@mail.xidian.edu.cn

[1]西安电子科技大学人工智能学院，中国西安 710071

[2]中国深圳鹏城实验室 518055

摘要

背景：随着高通量测序技术的飞速发展，全基因组测序成本迅速下降，导致基因组数据呈指数级增长。如何有效压缩大规模基因组项目产生的 DNA 数据，成为制约 DNA 测序产业进一步发展的重要因素。尽管近年来DNA碱基的压缩技术已经取得了显著的进步，但质量分数的压缩仍然是一项挑战。

结果本文通过重新研究质量得分与排序过程之间的内在关联，提出了一种基于自适应编码顺序（ACO）的新型无损质量得分计算方法。自适应编码顺序的主要目的是根据排序过程，以最相关的轨迹自适应地遍历质量分数。通过与自适应算术编码和改进的上下文策略合作，ACO 在下一代测序（NGS）数据中以中等复杂度实现了最先进的质量分数压缩性能。

结论ACO 已被 AVS（中国音视频编码标准工作组）采用，并可在 https://github.com/Yoniming/ACO 免费获取。

关键词高通量测序 质量得分压缩 无损压缩 自适应编码顺序

背景介绍

测序技术已逐渐成为广泛应用于生物研究的基础技术[1]。获取不同生物的遗传信息有助于我们增进对有机世界的了解。在过去的几十年中，人类全基因组测序（WGS）的价格已经降到了 1000 美元以下，其下降速度超过了摩尔定律的预期[2]。在这种情况下，下一代测序（NGS）数据的数量呈指数级增长，甚至超过了天文数据[3]。如何有效压缩大规模基因组项目产生的 DNA 数据，已成为制约 DNA 测序行业进一步发展的重要因素。因此，有必要对 NSG 数据进行压缩，以便于存储和传输。

具体来说，DNA 数据压缩有两个主要问题：核苷酸压缩和质量分数压缩。质量值占压缩数据的一半以上，已被证明比核苷酸数据更难压缩[4, 5]。为了提高总文件大小的压缩率，有必要对质量分数进行单独的效率改进。随着组装技术的发展[6]，核苷酸压缩技术取得了显著的进步，这使得质量分数压缩问题成为当前 DNA 数据存储和传输应用中的主要瓶颈之一。

质量得分（QS）代表下一代测序（NGS）数据测序过程中每个碱基字符的置信度，但字母更大（41-46 个不同等级）。现在有更多的仪器制造商，它们主要仍有大量离散的质量分数。研究质量分数的数据特征对压缩非常有帮助、
[7]揭示了相邻质量得分之间存在很强的相关性，这可视为当前无损质量得分压缩流水线的基础：（1）使用马尔可夫模型估计质量得分的条件概率；（2）通过光栅扫描顺序对读数的每个位置进行跟踪；（3）通过算术编码或范围编码对质量得分进行编码。

虽然已经提出了很多质量分数有损压缩方法[8-11]，但保留原始数据尤为重要。因此，我们将重点放在无损压缩上，最近有很多基于神经网络的压缩方法，将递归神经网络预测器与算术编码器相结合，对基因组数据集进行无损压缩[12, 13]。然而，不同测序机器产生的基因数据分布不同，这使得基于网络训练的方法需要对每种数据进行单独训练，不利于广泛应用。基于上述管道，人们提出了三种不同的无损压缩器 GTZ [14]、Quip [15] 和 FQZcomp [5]。这三种压缩器的区别仅在于马尔可夫模型阶数和上下文量化策略不同，因此压缩率约为 0.1%[16，17]，具体取决于数据分布情况。这就不可避免地提出了一种消极的观点，即进一步提高无损压缩率的空间不大。

本文通过对测序过程的重新研究，揭示了现有基于光栅扫描的质量分数压缩策略的两个主要缺点。首先，光栅扫描顺序是一种 "深度优先 "的读取遍历策略。然而，正如文献[7]所指出的，质量得分沿着单个读数呈下降趋势。这使得马尔可夫模型的片断静止假设站不住脚。其次，考虑到测序过程是通过二维多光谱成像进行的[18]，但 FASTQ 文件只是将质量得分存储到一维信号的堆栈中。基于栅格扫描的技术对每个读数进行独立压缩，无法探索空间相邻读数（而非 FASTQ 文件中的相邻读数）之间潜在的二维相关关系。

为了克服上述两个缺点，我们针对新一代测序（NGS）数据提出了一种基于自适应编码顺序（ACO）的新型质量分数压缩技术。与一般的压缩方法不同，ACO 是一种特殊的质量得分压缩器，因此它考虑了更多质量得分数据的分布特征。一般来

说，ACO 有三方面的贡献：（1）根据内部相关性对质量分数进行压缩；（2）对质量分数进行压缩；（3）对质量分数进行压缩。

(2) 根据排序原则将基础信息作为上下文信息，并量化复合上下文模型；(3) 采用蛇形编码序列。ACO 的主要目标是沿着最相对的方向遍历质量得分，这可以看作是将独立的一维质量得分向量堆重组为高度相关的二维矩阵。与现有技术相比，拟议 ACO 技术的另一项改进是复合上下文建模策略。正如我们将在 "方法 "一节中详细说明的那样，ACO 上下文模型由两个额外方面组成，而不是相邻的 QS 值：(1) 每个读数的全局平均值；(2) DNA 碱基的变异。复合上下文模型不仅有利于概率估计和算术编码，更重要的是，在实施过程中，它可以防止 ACO 对输入的 FASTQ 文件进行多次随机访问：压缩过程只需一条路径即可完成，但代价是上下文会被稀释并产生副信息。

实验结果表明，所提出的 ACO 技术在无损质量分数压缩方面达到了最先进的水平，与 FQZcomp 相比，压缩率提高了 5%以上[5]。ACO 的唯一缺点是内存成本，与 FQZcomp 相比，ACO 分别需要 *400M* 和 *4G* 的额外内存来缓冲二维质量得分矩阵和存储复合上下文模型，这对于目前的 PC 来说应该不再是一个大问题。

洞察力

本节将首先分析质量得分的数据特征，通过具体实例说明编码序列会对质量得分的计算产生一定的影响，从而促进我们沿着数据相关性最强的方向压缩质量得分。其次，我们通过分析 FASTQ 文件的测序原理和生成过程，挖掘质量得分数据中的额外相关性，从而建立一个新颖的复合上下文量化模型。这些分析结果构成了我们的创新成果，从而产生了轻便、可移植的质量得分压缩器 ACO。

编码顺序的影响

质量得分是对读数中相应核苷酸错误概率的估计，也是对碱基特征可靠性的评估。这些信息既可用于原始数据的质量控制，也可用于下游分析。我们在图 1 中给出了 ERR2438054 四个读数的质量得分分布情况，可以看出由于噪声（五角星标记的点）的影响，质量得分是一个随机的不稳定信号，相邻质量得分之间存在很强的相关性。因此，我们可以利用质量分数的这些特点来改变编码顺序，从而提高压缩比。改变顺序听起来并不像改变熵值，因为根据信息论，信源的信息量是概率的函数，用信源的信息熵来表示。但是，由于编码中使用了自适应算术编码器，编码器会定期更新符号概率，因此改变顺序可以减小比特流的大小。关于

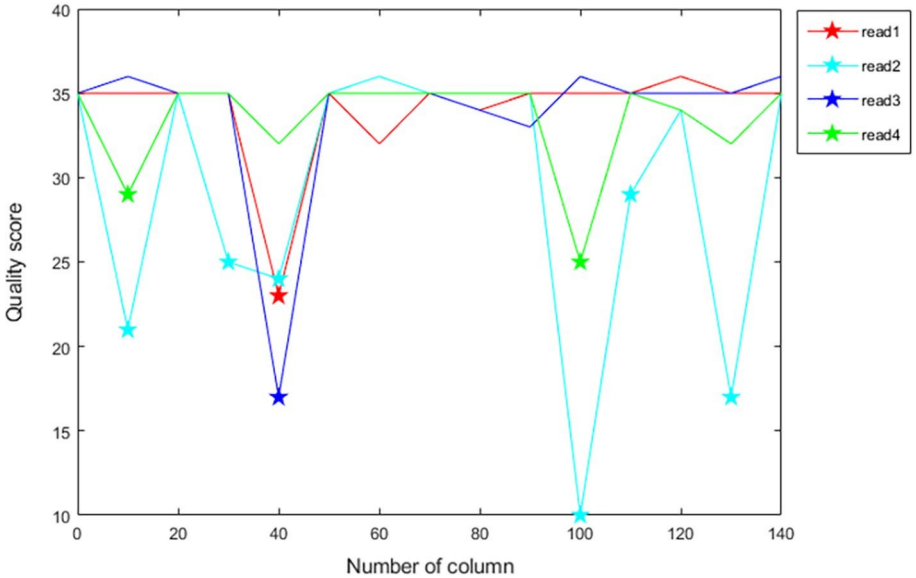

图 1 ERR2438054 的质量得分分布曲线

算术编码器的原理并不是本文的主要内容，因此我们只给出一个测试实验来说明编码顺序对压缩结果的影响。首先，我们创建两个随机信号 *X* 和 *Y*，假设随机信号 *X* 和 Y 是两个不同的高斯信号，并让 Z1 = X + Y ，Z1 表示 *X* 和 *Y* 是串联的，这并不意味着将相应的值相加。然后，随机打乱 Z1 的分布，记为 *Z2*，与两个高斯分布的 Z1 相比，洗牌后的 Z2 分布更稳定，再将 Z1 的分布按大小排序，记为 *Z3*。最后，三组不同的信号通过 0 阶算术编码器进行编码，比特流的结果为 S(Z3)< S(Z2)< S(Z1)（S(-)代表熵的大小）。这是因为排序过程相当于把分布相似、相关性强的数据放在一起。改变顺序后的编码能更好地配合自适应算术编码器的概率更新机制。

挖掘更多相关信息

以目前广泛使用的 Hiseq 测序平台为例，测序过程包括三个步骤：(1) 构建 DNA 文库；(2) 通过桥式 PCR 扩增生成 DNA 簇；(3) 测序。本文对测序步骤进行了研究，以挖掘更多质量得分之间的内在联系，从而帮助完成压缩任务。测序的基本原理基于流式细胞的多光谱成像。

如图 2 所示，测序过程包括五个步骤。第一步，在流动池中加入聚合酶和一种 dNTP，以激活特定簇的荧光。第 2 步，多光谱相机根据添加的 dNTP 拍摄流动池中特定波长的照片。然后，在步骤 3 中，采用化学试剂冲洗流动池，为下一次成像做准备。步骤 4

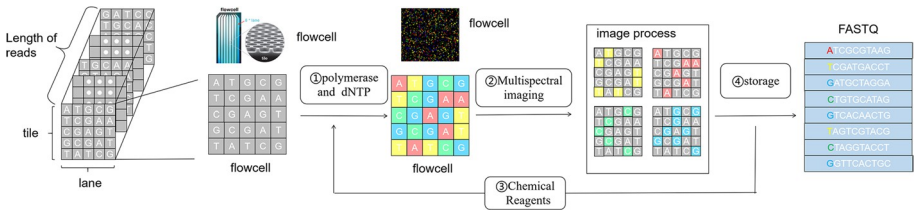

图 2 排序原理示意图

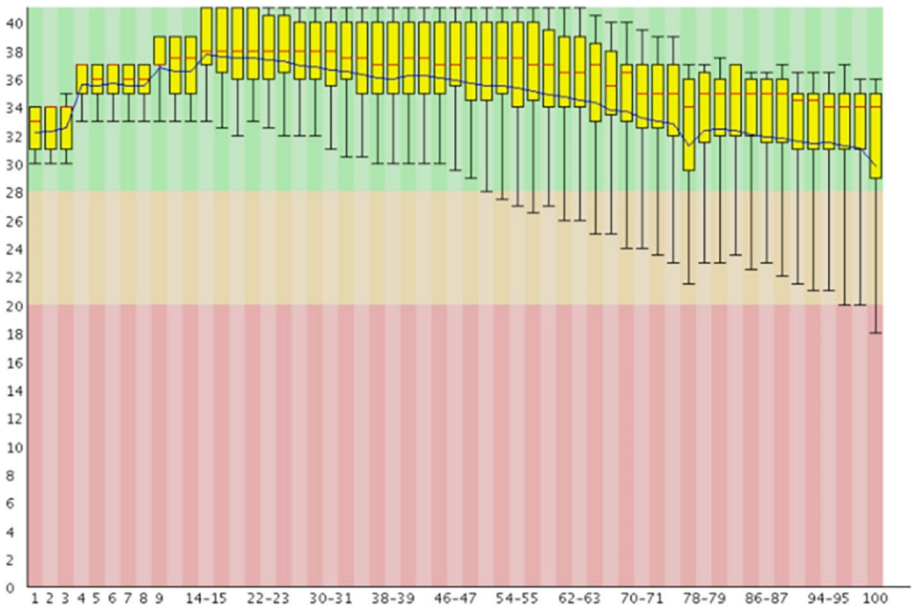

图 3 FASTQ 质量得分的分布情况

以上三个步骤重复四次，使用不同的 dNTP 和不同的成像波长，得到四通道多光谱图像。在第四步中，测序仪根据采集到的四通道图像，不仅能估计出每个簇最可能的类型，还能评估估计结果的可信度，并分别存储为碱基和质量得分。步骤 1-4 被视为一个测序周期，对流池中所有读数的一个位置（深度）进行测序。因此，在步骤 5 中，测序周期会重复多次，重复的周期数与读数的长度相对应。

正如我们在下文中详细讨论的，质量分值对应于三个方面：（1）循环次数；（2）基数变化；（3）芯片的局部位置。

循环次数会影响质量得分的分布。DNA 聚合酶用于合成和测序过程，在测序初期，合成反应不是很稳定，但酶的质量很好，所以会在高质量区域波动。随着测序的进行，反应趋于稳定，但酶的活性和特异性逐渐降低，累积误差逐渐放大。因此，出错概率增加，总体质量得分呈下降趋势。如图 3 所示（由

FastQC[19]），随着测序的进行，质量得分的平均值逐渐降低，而方差却在增加。因此，假定每个读数都是沿传统光栅扫描顺序的静态随机信号是不恰当的。

碱基变化也会影响质量得分的分布。正如我们之前所讨论的，流式细胞中碱基类型的识别是按照 dNTP 和波长的顺序进行四步循环的。例如，假设循环顺序为 "A-C-G-T"，如果一个读数的碱基是 "AA" ，则在第一个 "A "成像后，流式细胞要洗涤四次，直到第二个 "A "成像。但如果碱基是 "TA" ，机器只在 "T "成像前清洗一次流动池。这样，如果流式细胞簇中含有一些残留物，前一个 "A "碱基就会影响后一个 "T "碱基的成像过程，从而导致 "T "的模糊性，使 "T "的质量得分显著下降。虽然有些机器采用复合 dNTP 来代替四步循环，但残留物仍然会影响质量得分。因此，在压缩质量得分时，应将碱基变化作为一个侧面信息来模拟每个质量得分的边际概率。

芯片的局部位置会影响质量得分的分布。流室可视为一个二维阵列，每个簇对应阵列中的一个条目。如果一个入口的高振幅荧光浓度扩散到阵列的相邻入口，这就是著名的 "串扰 "现象[20]。换句话说，相邻质量得分之间存在二维空间相关性。然而，存储的 FASTQ 文件是所有读数的一维堆栈，忽略了二维相关性。因此，质量得分的压缩应该挖掘读数之间潜在的二维空间相关性。

方法

在本节中，我们将讨论所提出的基于自适应编码顺序（ACO）的质量分数压缩技术。ACO 的两个贡献是：(1) 使用自适应扫描顺序取代传统的光栅扫描顺序，后者形成的信号更加静止。(2) 使用复合上下文建模，在探索质量分数之间潜在的二维相关性的同时，考虑基数变化的影响。

沿着最相对的方向遍历质量得分

从图 3 中可以看出，随着读数长度的增加，列均值减小，但方差变大，这证明列与列之间存在很强的相关性。同时，列均值的减小也与具体测序的实际过程一致，即质量得分沿着单个读数呈下降趋势。由此可以验证，改变扫描顺序可以提高自适应算术编码器的性能，沿着更稳定的信号进行编码可以获得更好的编码效果。

所有基于算术编码器的压缩方法在遍历数据编码时都使用图 4a 中的扫描方法。在这种扫描方法下，质量分数是逐行编码的，扫描完一行后，下一步从第二行开始。显然，在对前一行的最后一个字符进行编码后，连接下一行的第一个字符会产生很

大的跳变，这种跳变会使信号之间的转换不稳定。因此，我们采用自适应扫描顺序来取代传统的 Ras- ter 扫描顺序，从而实现信号的稳定遍历，如图 4b 所示。

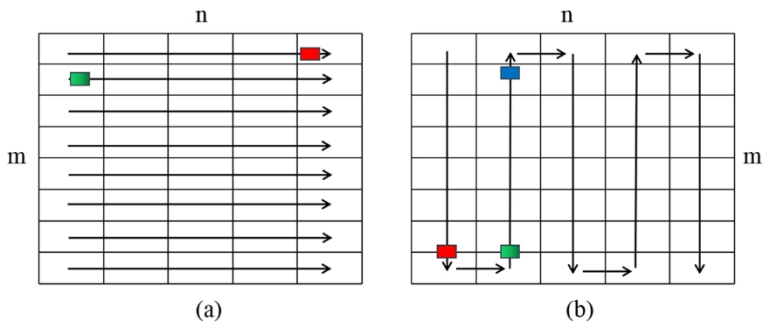

图 4 传统扫描与 ACO 扫描的比较：**a** 传统遍历方法；**b** 自适应扫描顺序

起点从第一个元素开始，沿列向下遍历，直到一列的末尾，然后从下一列的末尾向上遍历。与传统的扫描方式不同，ACO 采用的是一种类似蛇形的扫描方式。采用蛇形遍历的原因是为了使列与列之间的过渡更加平滑，一列的末尾符号与下一列的末尾符号更加相关，红绿符号之间的相关性明显强于红蓝符号之间的相关性。因此，在对红色符号进行编码后，选择绿色符号比蓝色符号更适合从第二列开始编码。通过改变扫描顺序，编码可以朝着更稳定的方向进行。在不引入其他因素的情况下，自适应算术编码器的概率更新机制得到了充分利用。

复合语境建模

正如声明部分所述，质量得分的压缩应挖掘读数之间潜在的二维空间相关性，因此我们使用复合上下文模型来表达质量得分数据中的额外相关性。ACO 上下文模型还包含两个方面，第一个方面是获取每个读数的全局平均值。根据《宣言》中的例子可以看出，调整数据顺序，使相似符号聚类在一起，会得到很好的压缩效果。如图 1 所示，四个读数的分布曲线非常相似，只有一些奇异点存在差异。因此，对具有相似行分布的数据进行聚类和编码是一种改进策略，但计算每一行的分布需要很多步骤，而且对相似行进行聚类也会带来时间和空间上的损失。我们计算每一行的平均值信息，以反映其分布情况，并对平均值相同的行进行分类。对于行信息来说，均值是静止性的衡量标准，均值相同的行可以近似地认为整行的分布基本相同，尽管一些奇异点可能会使分布曲线不完全重合。使用行均值可以节省大量的计算量和时间，而不需要计算行间的库尔贝-莱布勒离差，同时也不会浪费行间的相关性。行聚类法需要向解码器传输额外的信息来记录行序的变化，面临同样的问题，使用均值法也需要向解码器传输每行的均值信息。

解码器。在实际应用中，我们会比较额外的编码量和行均值带来的实际收益值。当收益大于原值时，我们会选择添加行均值信息作为上下文。

具体来说，在建立上下文模型的过程中，会出现上下文稀释的问题，因此我们需要设计一种合适的均值量化方法，从而解决上下文稀释的问题，这是一个动态编程问题，离散分布随机变量量化的优化目标是使失真最小。目标的表达式为

$$Ed(x, Q(x)) = \sum_{i} d(xi, Q(xi))p(xi) \tag{1}$$

其中 xi 是 x 的非零概率值，Q(xi) 是 xi 的量化值，$d(-, -)$ 是特定的失真度量。我们可以定义一个条件集 $M\theta$ = {mi, i = 1...N} 来表示每个特定的 mi 对应一个特定的 E 值。定义量化集 $Q\theta = \{qk, k = 1...K\}$，其中 K 代表量化变量、

所以 $M\theta = \bigcup_{k=1}^{K} qk, qi \quad qj = \varnothing$，如果 $i \neq j$。因此，每个子集 qk 对应于一个分区

的 mi，其 $p(qk) = \sum_{mj \in qk} p(mj).H(X|M)$ 和 $H(X|Q)$ 的表达式为

如下所示：

$$H(X|M) = \sum_{m_i \in M\, x_\theta} p(m_i) \sum p(x|m_i) log_2 \frac{1}{p(x|mi)} \tag{2}$$

$$H(X|Q) = \sum_{qk \in Q\, x_\theta} p(q_k) \sum p(x|q_k) log_2 \frac{1}{p(x|qk)} \tag{3}$$

可以看出，从 M 到 Q 生成的 qk 包含了所有的 mi，因此 $p(x|qk)$ 可以看作是 $p(x|mi)$ 的量化值。根据等式（1），我们可以将上下文量化为

$$L = \sum_{m_i \in M_\theta} d(p(x|mi), Q(p(x|mi)))p(mi) \tag{4}$$

其中 L 是量化目标函数，最小化 L 意味着至少获得局部最优的量化结果。上下文量化的优化目标变为：对于给定的量化级数 K（K < N），为 mi（i = 1，.....N）找到一个最佳分区方案，然后计算所有 K 分区的最佳量化值 $Q(p(x|mi))$，从而使等式（4）最小化。

通过计算动态程序设计问题，我们得到了适用于以下情况的结果

测试数据。为了提高实际压缩过程中的计算效率，我们将计算结果作为量化方法。如果需要提高指定文件的压缩率，用户可以单独求解优化问题，得到最佳的量化方法。最后，我们的行特性量化方法如图 5 所示：

可以看出，这种情况下的量化方法与加入阈值的有损压缩非常相似。不同之处在于，我们对行

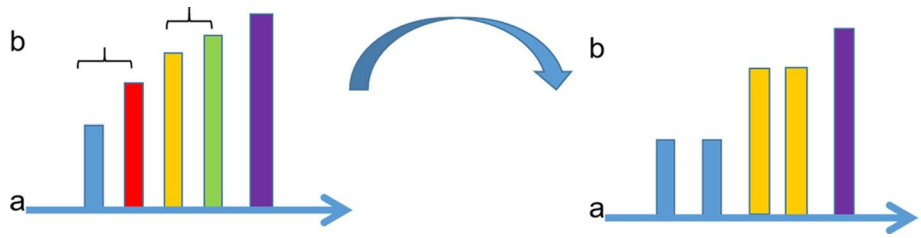

图 5 行平均量化方式

在不影响无损解码的情况下，只需提取各行之间的相关特性，并使用动态编程的方法就能得到更好的结果。包括当前值 q 的最终定量方法是

$$\begin{aligned}
&\bullet \textit{ 如果 } (q < 30) && \textit{则 } q = 30\\
&\bullet \textit{ 否则，如果 } (30 \le q < 32) && \textit{则 } q = 32\\
&\bullet \textit{ 否则，如果 } (32 \le q < 34) && \textit{则 } q = 34\\
&\bullet \textit{ else if } (34 \le q < 36) && \textit{则 } q = 36\\
&\bullet \textit{ else if } (36 \le q < 38) && \textit{则 } q = 38\\
&\bullet \textit{ 否则} && q = q
\end{aligned} \tag{5}$$

另一方面，质量分数的分布是随机的，波形不会受质量分数值的影响而平滑过渡。因此，与波形模式不同，我们可以从这些奇异点的排序原理入手。

[21] 发现相邻碱基之间的质量得分通常是相似的，质量得分的概率分布受碱基分布的影响。考虑到核苷酸序列在获取过程中存在一定的天然相似性，碱基分布被视为衡量质量得分是否存在正弦点的标准，用于模拟两个符号之间的静止性。碱基阶数的增加会使模型呈指数增长，在平衡模型大小和效果提高率的情况下，我们选择二阶来描述碱基与质量得分之间的相关性。在 FASTQ 文件中，一个质量得分对应一个碱基，条件熵为：

$$-\log_2 \prod_{i=1}^{n} p(x_i|j_i, j_{i-1}) < -\log_2 \prod_{i=1}^{n} p(x_i|j_i) \tag{6}$$

其中，j_i 为当前代码质量得分 x_i 的基值，该公式表明了基值对熵的影响。综合所有上下文模型后，我们提供了最终的复合上下文建模策略（如图 6 所示）和 ACO 算法（如算法 1 所示）。

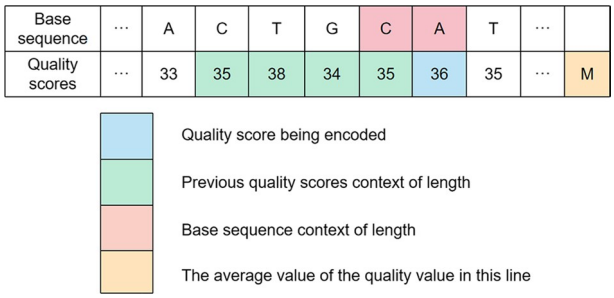

图 6 综合语境建模策略

Algorithm 1:ACO algorithm framework

Input: A FASTQ file.
Output: The compressed quality score file
STEP1: Data preprocessing
1.Use a $N\times K$ matrix **Q** to store the quality score of FASTQ file.
2.Use a $N\times K$ matrix **P** to store the base value of FASTQ file.
for all $0 \le n \le N$ **do**
for all $0 \le k \le K$ **do**
calculate the max of **Q**(n,k) as symbol_max
calculate the min of **Q**(n,k) as symbol_min
symbol_number=symbol_max-symbol_min+1
end for
end for
STEP2: Composite context model
1.Calculate model_num by $model\_num = symbol\_number^3$.
2.Use a $N\times 1$ vector **M** to store the mean value of the quality value in each line.
for all $0 \le n \le N$ **do**
for all $0 \le k \le K$ **do**
$A = \mathbf{M}(n, 1);$
$B = max[\mathbf{Q}(n, k-1), \mathbf{Q}(n, k-2)];$
$C = max[\mathbf{Q}(n, k-3), \mathbf{Q}(n, k-4)];$
$if \quad ([\mathbf{Q}(n, k-3) == \mathbf{Q}(n, k-4)]) \qquad D = 0;$
$else \qquad D = 1;$
$E = [\mathbf{P}(n, k), \mathbf{P}(n, k-1)];$
$model\_idx = [A, B, C, D, E]$
end for
end for
STEP3: Snake traversal coding
i=0;
for all $0 \le k \le K$ **do**
for all $0 \le n \le N$ **do**
$if \quad (k\%2 == 0) \qquad i = k;$
$else \qquad i = N - n - 1;$
use arithmetic encoder to compress **Q**(n,k) with model_idx
end for
end for

结果和讨论

在本节中，我们将 ACO 算法的性能与其他先进算法进行了比较，并报告了结果。我们将我们的算法与 gzip 和 7zip 等通用压缩算法以及 GTZ、fqzcomp 和 quip 等特定领域的算法进行了比较。我们的重点仅限于无损压缩，而没有评估一些有前途的有损方法，也没有评估仅能压缩核苷酸序列的方法。对于 fqzcomp 算法，我们比较了 q1、q2 和 q3 压缩模式的结果；对于 GTZ 算法，它没有单独显示质量分数压缩结果，因此我们比较了正常模式。值得注意的是，ACO 在压缩对齐质量得分方面更具优势，而且不接受原始 FASTQ 文件以外的任何输入。同时，我们也不将接受任何参考基因组的算法纳入比较范围。实验中使用的数据集是从美国国家生物技术信息中心-序列读取档案（NCBI- SRA）数据库[22]下载的 FASTQ 格式文件，如表 1 所示。

所有实验都在一台配备 Inter Core i9-9900K CPU 3.60GHz 处理器、32GB 内存、2TB 磁盘空间和 Ubuntu 16.04 的服务器上进行。所有算法都以压缩率（CR）和比特/质量值（BPQ）进行比较。CR 和 BPQ 的定义如下：

$$CR = \frac{Lafter}{开始} \times 100\% \tag{7}$$

$$BPQ = 8 * CR \tag{8}$$

其中 $Lafter$ 表示压缩后的文件大小，$Lbegin$ 表示压缩前的文件大小，表 2 总结了所有算法在 NGS 数据集上的压缩结果。

表 2 给出了 ACO 相对于每种比较算法的改进，并通过压缩率进一步反映了我们所使用方法的优势。表 2 中的最佳结果已用粗体标出。与 Gzip 相比，文件大小平均减少了 32.01%，与最佳设置下的 7-Zip 相比，文件大小平均减少了 32.93%。结果表明，所提出的 ACO 算法在六个代表性数据上取得了更好的效果。特别是，ACO 获得了 27.67% 的平均压缩率，使质量分数数据的 大小减少了 72.33% 以上。同时，平均 2.21 的 BPQ 结果要小得多

表 1 用于评估的 6 个 FASTQ 数据集说明

运行标识	测序平台	FASTQ 大小（字节）	阅读长度	质量大小（字节）
NA12878_2	BGISEQ-500	134363357648	2*100	56983386200
ERR2438054_1	BGISEQ-500	133406591610	2*150	47097570000
ERR174324_1	Illumina HiSeq 2000	57800970448	2*101	22580690796

ERR174328_1	Illumina HiSeq 2000	57210954538	2*101	22350322320
ERR174327_1	Illumina HiSeq 2000	54724344869	2*101	21379957043
ERR174324_2	Illumina HiSeq 2000	57800970448	2*101	22580690796

表 2 NGS 数据集的所有算法压缩结果

运行标识	比率	压缩	7z	gtz	调侃	fqz-q1	fqz-q2	fqz-q3	春季	ACO
NA12878_2	总登记率(%)	48.55	49.66	38.47	38.48	39.08	38.59	38.35	39.68	**36.38**
	BPQ	3.88	3.97	3.08	3.08	3.13	3.09	3.07	3.17	**2.91**
ERR2438054_1	总登记率(%)	46.23	47.11	37.09	36.52	37.08	36.71	37.63	37.07	**34.55**
	BPQ	3.70	3.77	2.97	2.92	2.97	2.94	2.92	3.01	**2.76**
ERR174324_1	总登记率(%)	36.58	36.94	25.47	26.14	27.30	25.81	24.90	26.39	**23.86**
	BPQ	2.93	2.96	2.04	2.09	2.18	2.06	1.99	2.11	**1.91**
ERR174331_1	总登记率(%)	36.55	36.91	25.45	26.11	27.27	25.77	24.86	26.37	**23.87**
	BPQ	2.92	2.95	2.04	2.09	2.18	2.06	1.99	2.11	**1.91**
ERR174327_1	总登记率(%)	35.53	35.88	24.56	25.31	26.39	24.90	24.02	25.45	**22.97**
	BPQ	2.84	2.87	1.96	2.02	2.11	1.99	1.92	2.04	**1.84**
ERR174324_2	总登记率(%)	38.47	38.81	27.07	27.52	28.89	27.37	26.35	28.03	**25.52**
	BPQ	3.08	3.10	2.17	2.20	2.31	2.19	2.11	2.24	**2.04**

比 ASCII 格式的原始 8 BPQ 要高。两个评价标准表明，ACO 算法对同一文档的不同处理方法取得了最佳的压缩效果。根据平台的不同，本文提出的 ACO 算法设置了不同的模式和处理策略，使得压缩效率更高。

结论和未来工作

本文介绍了基于自适应编码顺序的无损质量分数压缩算法 ACO。ACO 沿着最相对的方向遍历质量分数，并使用复合上下文建模策略来实现最先进的无损压缩性能。不过，目前的 ACO 版本，特别是提出的复合上下文建模策略，是针对第二代测序机器提出的。对于第三代测序数据，可以根据基因组质量得分修改复合上下文模型，但随着上下文模型的增加，可能会出现上下文稀释问题。另一种解决方案是使用深度学习技术来估计每个质量得分的边际概率，以取代当前的上下文建模。在下一步工作中，我们将集中研究上述两种策略，并将 ACO 扩展到第三代测序数据。此外，如何针对不同数据选择最佳方式，自动计算平均值作为上下文信息，并将其应用于仅包含 8 个质量值的数据，也将是我们下一步的研究工作。

缩略语

NCBIN 国家生物技术信息中心 NGS 下一代测序

WGS全 基因组测序 QS质量得分

CRC 压缩比

每个质量值 的 BPQ 位数

致谢

我们要感谢编辑和审稿人对这项工作提出的宝贵意见，这些意见有助于提高本文的质量。

作者供稿

YN和MM构思了算法、开发了程序并撰写了手稿。FL和GS帮助编辑脚本，设计并进行实验。XL 准备了数据集，进行了分析并帮助设计了程序。所有作者都阅读并批准了最终手稿。

资金筹措

本研究得到国家重点研发计划（2019YFE0109600）、国家自然科学基金委（No.61875157、61672404、61632019、61751310和61836008）、国防基础科学研究计划（JCKY2017204B102）、西安市科技计划（20191122015KYPT011JC013）、中央高校科研基金（RW200141、JC1904和JX18001）、国家重点研 发 计 划（ 2018YFB2202400）的部分资助。

数据和材料的可用性

项目名称：ACO。项目网站： https://github.com/Yoniming/ACO。操作系统：Linux 或 Windows：Linux 或 Windows。编程语言：C/C++：C/C++。其他要求：GCC 编译器和存档工具 "tar"。许可证MIT 许可。对非学术界人士的使用有任何限制：如需用于商业用途，请联系作者。所有数据集均从 NCBI 的 SRA 下载。支持本文结论的所有数据均包含在文章及其附加文件中。

声明

伦理批准和参与同意书

不适用。

同意出版

不适用。

竞争利益

作者声明他们没有利益冲突。

收到：接受： 2022 年 4 月 25 日
Published online: 07 June 2022

参考资料

1. You Z-H, Yin Z, Han K, Huang D-S, Zhou X. A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network.Bmc Bioinform.2010;11(1):343.
2. Wetterstrand KA.DNA 测序成本：来自 NHGRI 基因组测序计划 (GSP) 的数据。www.genome.gov/sequencingcostsdata (2016).
3. Stephens ZD.大数据：天文还是基因？Plos Biol.
4. Ochoa I, Hernaez M, Goldfeder R, Weissman T, Ashley E. 有损压缩质量分数对变体 调用的影响。Brief Bioinform.2016;18(2):183-94.
5. Bonfield JK, Mahoney MV.fastq 和 sam 格式测序数据的压缩。PloS One.2013;8(3):59190.
6. Bromage AJ.组装大型基因组的简洁数据结构。Bioinformatics.2011;27(4):479-86.
7. Kozanitis C, Saunders C, Kruglyak S, Bafna V, Varghese G. 使用 slimgene 压缩基因组序列片段。J Comput Biol.
8. Rodrigo C, Alistair M, Andrew T. 基因组数据质量分数的有损压缩。生物信息学》。2014;15:2130-6.
9. Greg M, Mikel H, Idoia O, Rao M, Karthik G, Tsachy W. Qvz：质量值的有损压缩。Bioinformatics.2015;31:3122-9.
10. Bonfield JK, McCarthy SA, Durbin R. Crumble: reference free lossy compression of sequence quality values.Bioinformatics.2018;35(2):337-9.
11. Shibuya Y, Comin M. 通过基于序列的质量平滑更好地压缩质量得分。BMC Bioin- form.2019;20-S(9):302:1–11.
12. Mohit G、Kedar T、Shubham C、Idoia O. DeepZip：使用递归神经网络的无损数据压缩。2019 数据压缩大会（DCC），2019 年，第 575 页。
13. Shubham C, Kedar T, Wen C, Wang L. LFZip：通过 改进预测对多元浮点时间序列数据进行有损压缩。2020数

据压缩大会（DCC），2020年，第342-51页

14. Xing Y, Li G, Wang Z, Feng B, Song Z, Wu C. Gtz: 针对 fastq 文件优化的快速压缩和云传输工具。BMC Bioinform.2017;18(16):549.
15. Jones DC, Ruzzo WL, Peng X, Katze MG. 高效从头组装辅助压缩下一代测序读数。Nucleic Acids Res. 2012; 40(22):171-171.
16. Shubham C, Tatwawadi K, Ochoa I, Hernaez M, Weissman T. Spring: a next-generation compressor for fastq data.生物信息学。2019;35:2674-6.
17. Yami SA, Huang CH.Lfastqc：无损的非基于参考的 fastq 压缩器。PLoS ONE.2019;14(11):0224806.

18. Sanger F, Nicklen S, Coulson AR.使用链终止抑制剂进行 DNA 测序。Proc Natl Acad Sci. 1977;74(12):5463-7.
19. Murphy TI.LaTeX 文档中的行距。[Online]. https://www.bioinformatics.babraham.ac.uk/projects/ fastqc/.
20. Geiger B, Bershadsky A, Pankov R, Yamada KM.细胞外基质与 细胞骨架之间的跨膜串联。Nat Rev Mol Cell Biol. 2001; 2(11):793-805.
21. Das S, Vikalo H. 通过维特比算法为illumina的下一代dna测序系统进行碱基划分。In: 2011 49th annual allerton conference on communication, control, and computing (Allerton). IEEE, pp.IEEE, pp.
22. Leinonen R, Sugawara H. 国际核苷酸序列数据库（2010 年）。

出版商说明