Lab1 寻找样本 DNA 序列中的重复片段

2025-02-24

**1 项目背景**

**1.1 DNA 基础知识**

DNA 是生物的遗传物质，主要由四种碱基组成：腺嘌呤（A）、胸腺嘧啶（T）、胞 嘧啶（C）和鸟嘌呤（G）。其中， A 与 T 形成碱基配对， C 与 G 形成碱基配对， 这种 互补配对由氢键维持。

DNA 分子是双链结构， 由两条互补的链以双螺旋方式缠绕在一起。由于碱基互 补配对的特性，DNA 的两条链是反向的（即 5,→3, 方向与 3,→5, 方向相对， 可参 考[3.3](#bookmark1)节例图）。例如，若序列为 5,-ATCG-3,，其反向互补序列为 3,-TAGC-5,，或者写 作 5,-CGAT-3, （一般情况下默认 5, 写在左侧，3, 写在右侧）。这就是说，序列 ATCG 和 CGAT 是反向互补的。

**1.2 测序与比对**

DNA 测序是一项基因组学中的基础技术，可以确定 DNA 分子中碱基（A 、T、C、 G）的排列顺序。人类基因组大约由 30 亿个碱基对组成，无法一次性测序整个基因组， 因此目前常用的第二代测序技术通常会生成数百万条短 DNA 片段。为了确定这些短序 列在基因组中的确切位置，必须将它们与参考基因组进行比对。

**1.3 DNA 序列比对的挑战**

将短读段（query）与参考基因组（reference）进行比对是一个计算上的挑战，主要 困难包括：

• 庞大的基因组规模: 参考基因组非常庞大（数十亿碱基对）

• 突变和变异: DNA 片段可能有测序错误或突变，因此需要使用模糊匹配

• 反向互补序列: DNA 是双链结构，在比对时需要考虑原始序列及其反向互补序列

**2 项目目标**

**2.1 整体目标**

本学期项目的最终目标为设计一个将短读段（query） 序列比对到参考基因组 （reference）的算法，需要同时兼顾算法的时空复杂度和最终结果的准确性。由于 DNA 中的变异种类繁多，每个 Lab 会重点关注其中一种或多种变异类型。同学们需要在各 次 Lab 中不断完善代码，使之能应对复杂的变异情况。

**2.2 本次** **Lab 目标**

Lab1 的目标是初步实现一个较为简单的比对算法， 主要考虑“重复”这一种变异类 型，暂时不考虑其他变异。要求在 PPT 给出的 query 和 reference 中，找到 query 中 存在的重复部分， 并输出其相关信息（所在 reference 位置、重复片段长度、重复数量、 是否为反向重复）。

**3 实现细节**

**3.1 重复变异**

设 Squery 和 Sreference 是两个由 ATCG 构成的字符串，分别表示短读段和参考基因 组。本次 Lab 中考虑的重复变异可以定义如下：

**正向重复**

正向重复变异描述的是， 某个序列 s 在 reference 的某个位置仅出现一次， 但在 query **对应位置**中连续出现了多次。即：

Sreference = —S1 sS2 —

Squery = —S1 sS3 ss · · · ssS2 —

其中：

• S1 , S2 是重复变异上下游的片段

• S3 一般为空，在本次 Lab 中还可能是构成另一组重复的相关序列（详见[3.3](#bookmark1)节）

• 重复片段 s 在 reference 和 query 中完全一致，**暂时**不考虑序列 s 内部出现的其 他变异

**反向重复**

反向重复变异描述的是， 某个序列 s 在 reference 的某个位置仅出现一次，但其反向 互补序列 s′ 在 query **对应位置**中连续出现了多次。即：

Sreference = —S1 sS2 —

Squery = —S1 sS3 s′ s′ · · · s′ s′ S2 —

其中：

• S1 , S2 是重复变异上下游的片段

• S3 一般为空，在本次 Lab 中还可能是构成另一组重复的相关序列（详见[3.3](#bookmark1)节）

• 重复片段 s 和 s′ 完全符合反向互补的规则，**暂时**不考虑序列 s′ 内部出现的其他 变异

• 本次 Lab 中默认 query 中第一次出现的重复片段（即 S1 , S3 之间的 s）是正向的， 这是因为这一情况在生物体系中更常见。但不排除实际数据中存在重复片段第一 次出现就是反向的情况，即 —S1 s′ S3 s′ s′ · · · s′ s′ S2 —

**重复数量**

重复数量指的是重复片段在 query 中**额外**出现的重复次数。由于重复片段 s 必须在 reference 中出现一次，我们在计算重复数量时需要扣除 query 中与 reference 对应的这 一次重复。即对于如下输入：

Sreference = —S1 sS2 —

Squery = —S1 sssS2 —

重复片段 s 在 reference 该位置出现 1 次，在 query 对应位置出现 3 次，相比于 reference 额外出现 2 次。故认为重复数量为 2。

**3.2 输入与输出**

**代码输入** ：query 和 reference 两个序列，均为由 ATCG 构成的字符串。详见 PPT 第 30-31 页。

**代码输出**：对于每一组重复，需要输出以下信息（可参考 PPT 第 29 页）

1. 第一次**额外重复**在 reference 的位置（注意： 重复片段的第一次出现应当是与 reference 相对应的，需要输出的是额外重复部分开始位置）

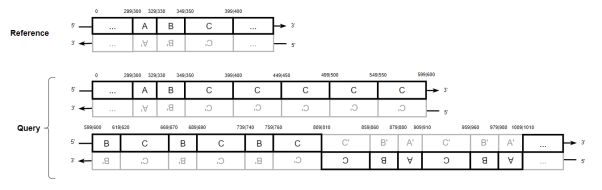
2. 重复片段的长度

3. 重复数量（注意：重复数量是指**额外**出现的次数）

4. 是正向重复还是反向重复

**3.3 样例分析**

以本次 Lab 的 query 和 reference 为例，作图如下。上方的数字是字符串下标。



其中，A′ , B′ , C′ 分别是序列 A, B, C 的反向互补序列。由于 DNA 双链之间的互补配对 关系，我们一般只记录其中的一条链，例如 PPT 给出的 DNA 序列只记录了上面这条 链，用符号可以表示为：

Sreference = · · · ABC · · ·

Squery = · · · ABC-CCCC-BCBCBC-C′ B′ A′ C′ B′ A′ · · ·

得到的结果表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 重复片段 | reference 中 的出现位置 | 序列长度 | 重复数量 | 是否反向 |
| 1 | C | 400 | 50 | 4 | False |
| 2 | BC | 400 | 70 | 3 | False |
| 3 | ABC | 400 | 100 | 2 | True |

**该表的解释如下：**

对于重复片段 C 而言：

• 其在 query 中第一次出现是在 350-399，与 reference 一致

• 其在 query 中第二次出现（即第一次额外出现重复） 是在 query 的 400-449，相当 于在 reference 的 400 位置处增加了序列 C，故位置为 400

• 序列长度为 C 序列自身长度 50

• 重复数量是 C 序列额外连续重复的次数， 即 query 中的 400-449, 450-499, 500-549, 550-599，共 4 次

对于重复片段 BC 而言：

• 其在 query 中第一次出现是在 330-399，与 reference 一致

• 其在 query 中第二次出现（即第一次额外出现重复） 是在 query 的 600-669，但与 reference 对比，仍然相当于在 reference 的 400 位置处增加了序列 BC，故位置仍 为 400

• 序列长度为 BC 序列长度 70

• 重复数量是 BC 序列额外连续重复的次数， 即 query 中的 600-669, 670-739. 740-809，共 3 次

对于重复片段 ABC 而言：

• 其在 query 中第一次出现是在 300-399，与 reference 一致

• 其在 query 中第二次出现（即第一次额外出现重复） 是在 query 的 810-909，但与 reference 对比，仍然相当于在 reference 的 400 位置处增加了序列 ABC 的反向互 补序列，故位置仍为 400

• 序列长度为 ABC 序列长度 100

• 重复数量是 ABC 序列额外连续重复的次数， 即 query 中的 810-909, 910-1009，共 2 次

**4 相关要求**

**需要完成的任务：**编写代码，根据输入的 query 和 reference 计算 query 中存在的重复 片段，并输出相关信息（详见[3.2](#bookmark2)节）。编程语言不限。

**结果要求：**本次 Lab 只需要考虑常见的重复变异， 能正确处理 PPT 所给样例并给出正 确结果即可。

**扩展性要求：**本次 Lab 无需考虑其他复杂的情况。但请注意， 后续 Lab 将在本 Lab 基 础上修改，请适度注意代码的可扩展性。

**复杂度要求：** 时间复杂度不得高于平方量级（不含平方量级）。

**提交要求：**在 Elearning 上提交实验文档，包括算法伪代码、时空复杂度、运行结果截 图。在 GitHub 上创建仓库（Repository）并上传代码。

**截止时间：** **2025.3.30 23:59**

**5 常见问题**

1. 哪些变换是合法的

简单而言，**在本次** **Lab 中**，只有上述提到的两类变换（正向重复、反向重复） 是 合法的。以下变换在本 Lab 中均不合法，并附上了不合法的原因。

(1) 重复部分有正向也有反向，例如：

Sreference = · · · AAA · · ·

Squery = · · · AAATTTAAAAAA · · ·

不能简单认为这是序列 AAA 在 query 中出现了 3 次额外重复。这是因为在 生物体系中，这种同时出现正向和反向重复片段的复杂情况往往包含了其他 变异类型，无法通过简单的重复变异解释。

(2) 重复片段自身发生变化，例如：

Sreference = · · · CAAT · · ·

Squery = · · · CAAAAAAAAT · · ·

正确理解是序列 AA 在 query 中出现了 3 次额外重复， 不能理解为 AAT 发 生重复。这是因为目前要求 query 中的重复片段与 reference 中的序列完全 一致（或反向互补）

2. 重复片段 s 的长度有无限制

理论上没有限制。不过， 在后续 Lab 中，如果重复片段太短（例如只有 1 个碱基 对），很可能会被当做其他变异而非重复变异。

3. pos 有多个解的情况如何处理

重复片段所在位置（pos）的数值无需特别精确， 例如在重复片段与上下游序列相 似的输入中：

Sreference = · · · ATCGATAT · · ·

Squery = · · · ATCGATCGATAT · · ·

可观察到两种重复的方式：

Sreference = · · · ATCG-ATAT · · ·

(Alignment A)

Squery = · · · ATCG-ATCG-ATAT · · ·

或

Sreference = · · · AT-CGAT-AT · · ·

(Alignment B)

Squery = · · · AT-CGAT-CGAT-AT · · ·

实际上，这两种结果对应的 pos 差异很小（只相差 2 个碱基对），这一误差在整个 基因组的 30 亿碱基对中可以忽略不计。在提交结果时，只需选择其中一个即可。 **注意：**本次 Lab 不涉及该问题，理论上可以得到唯一精确的 pos。

4. 比对为什么要用到 reference，直接看 query 中重复出现的片段是否可以

以上关于重复变异的介绍存在一个重要前提， 即 query 中发生重复的位置是与 reference 对应的，因此不能在脱离 reference 的情况下讨论重复变异。例如：

Sreference = —S1 sS2 S3 S4 —

Squery = —S1 sS2 S3 ssssS4 —

这种情况不属于重复变异， 因为 reference 的 S3 和 S4 之间不存在序列 s ，query 中多出的 ssss 序列属于插入变异， 而不是重复变异。更形象的比喻是， 重复变异 是将一段序列**原地复制**若干次，且不会影响序列其他位置。

5. 什么是“最大化重复片段长度、最小化重复次数”

在下面这一样例中：

Sreference = · · · CGATATCG · · ·

Squery = · · · CGATATATATATATCG · · ·

存在两种解释，即

Sreference = · · · CGAT-AT-CG · · ·

(Alignment A)

Squery = · · · CGAT-AT-AT-AT-AT-AT-CG · · ·

或

Sreference = · · · CG-ATAT-CG · · ·

(Alignment B)

Squery = · · · CG-ATAT-ATAT-ATAT-CG · · ·

其中，Alignment A 中的重复片段为“AT”，重复次数为 4 次。Aligment B 中的 重复片段为“ATAT”，重复次数为 2 次。“最大化重复片段长度、最小化重复次 数”主要是针对这类情况的要求， 即**同一位置**存在多种可能的重复片段时，找到 尽可能长的重复片段，从而降低重复次数。而对于不同位置的重复片段，其长度 和重复次数没有可比性。

**6 重要提醒**

1. 下一个 Lab 将在本次 Lab 基础上拓展，并考虑更多变异类型。请适度注意代码的 可扩展性，尽量减少返工

2. 下周一（3.10）习题课可能会提供一些相关思路，建议在课前先自行思考