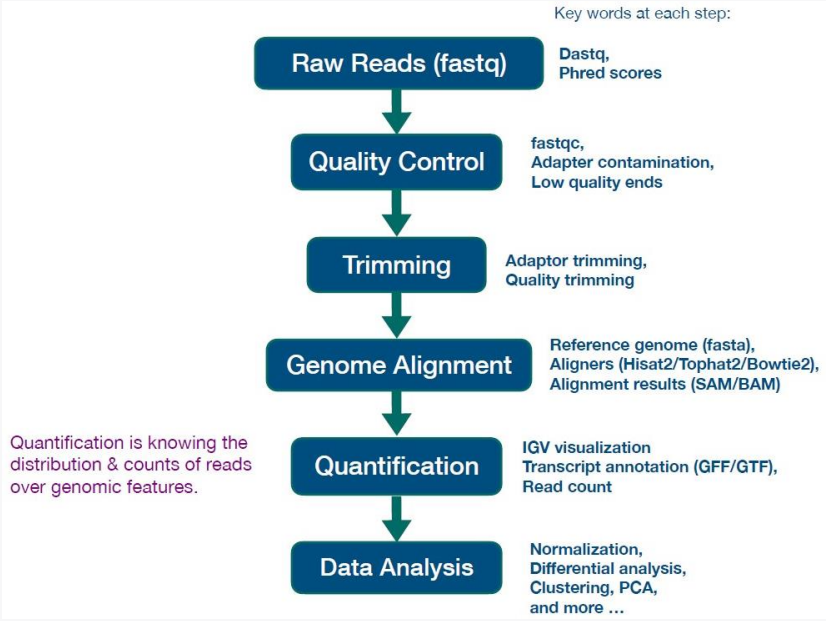
An overview of the Chinese version [4]

Overview of NGS pipeline

NGS（Next-Generation Sequencing，下一代测序）数据分析通常需要经过多个步骤，这些步骤可以通过 NGS 流程（pipeline）自动化执行。NGS pipeline 可以大大简化数据分析过程，并提供了高度可重复性和可比性。下面是一个常见的 NGS 数据分析流程：

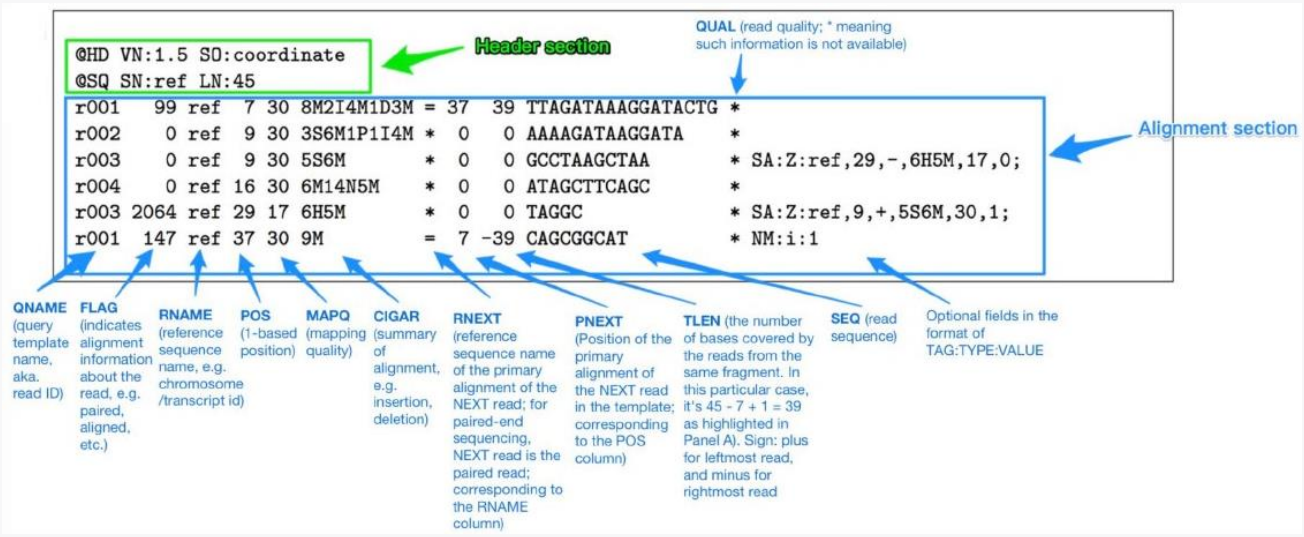


1. Quality Control（质量控制）：首先需要对原始测序数据进行质量控制，以确保后续分析的可靠性和准确性。常见的质控工具包括 FastQC 和 MultiQC。
2. Trimming（修剪）：在质控过程中，我们可以发现一些质量较低的序列。为了避免这些序列对后续分析的影响，需要对原始序列进行修剪（trimming）。修剪通常是在 read 的两端截断，去除低质量碱基、接头污染和重复序列等。常见的修剪工具包括 Trim Galore、Trimmomatic 和 Cutadapt。
3. Mapping（比对）：修剪过的序列需要与参考基因组进行比对，以确定每个 read 的位置和方向。常用的比对工具包括 Bowtie2、BWA 和 STAR。
4. Quantification（定量）：比对后，需要对每个基因或转录本的表达量进行计算。一些流行的工具包括 featureCounts、Kallisto 和 Salmon。
5. Differential Expression Analysis（差异表达分析）：对于 RNA-seq 数据，通常需要进行差异表达分析，以确定基因或转录本在不同条件下的表达差异。常用的差异表达分析工具包括 DESeq2、edgeR 和 limma-voom。
6. Functional Analysis（功能分析）：最后，对差异表达的基因或转录本进行功能分析，以确定哪些生物学过程、通路和功能受到调节。常用的功能分析工具包括 GOseq、KEGG 和 GSEA。

SAM format

SAM（Sequence Alignment/Map format）是一种文本格式通常用于存储测序数据的比对结果，可以被许多测序分析工具（如 SAMtools、Picard、GATK 等）所使用。SAM 格式的缺点是文件体积较大，而且读取速度较慢。因此，在实际应用中，通常会将 SAM 格式转换为 BAM（二进制 SAM）格式，以便更快速地处理和存储数据。

SAM 格式由多行文本组成，每行代表一个测序 read 的比对结果。每行中的列用制表符分隔，列中存储了 read 的相关信息，包括：



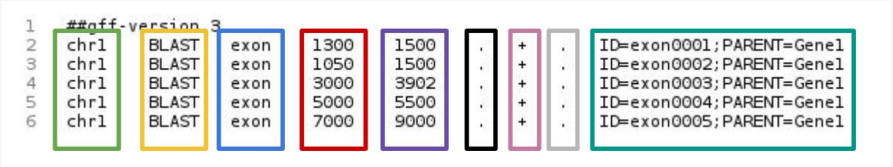
1. QNAME：read 的名字
2. FLAG：read 的状态信息，如是否匹配、是否在参考基因组上正向或反向、是否被过滤等
3. RNAME：比对到的染色体或参考序列的名字
4. POS：起始比对位置，如果比对到反向链，则是比对终止位置
5. MAPQ：比对质量分数，代表比对的可靠程度
6. CIGAR：描述比对的匹配情况，如匹配的长度和位置、插入和删除的碱基等
7. RNEXT：下一个比对的序列的名字
8. PNEXT：下一个比对的序列的起始位置
9. TLEN：read 的长度，包括所有碱基和可能的插入和删除

此外，SAM 格式还可以包含一些可选字段，如 read 序列、序列质量等信息。

GTF/GFF format

GTF（Gene Transfer Format）和 GFF（General Feature Format）是两种常见的基因组注释格式，用于描述基因组上的转录本和注释信息。基因组注释是指比现在更早进行的基因组实验，常用的基因组注释是基因、转录物、外显子、内含子、CDS 和各种表观遗传标记。

GTF/GFF 格式通常由多列组成，每列包含一些特定的注释信息。以下是 GTF/GFF 格式中的一些常见列：



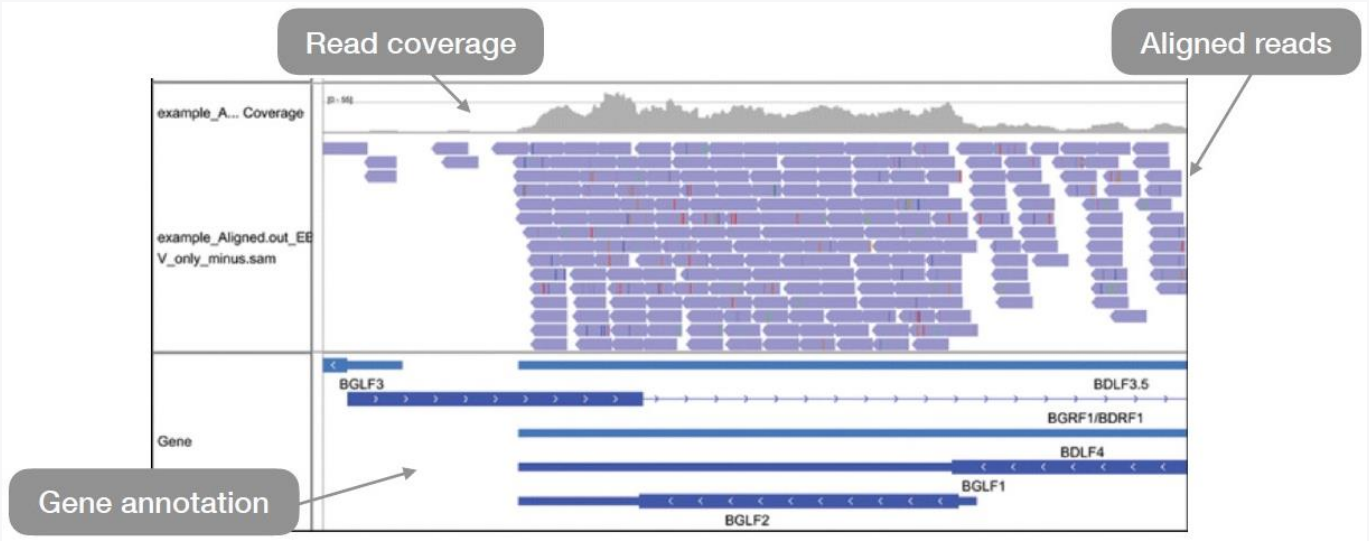
1. Sequence ID：染色体或序列的 ID
2. Source：注释数据来源，如 Ensembl、RefSeq 等
3. Feature type：特征类型，如基因、转录本、外显子等
4. Start position：特征起始位置
5. End position：特征结束位置
6. Score：特征得分
7. Strand：链向（+或-）
8. Frame：在 CDS（Coding Sequence）特征类型下，该列表示第一个核苷酸的相对位置

IGV

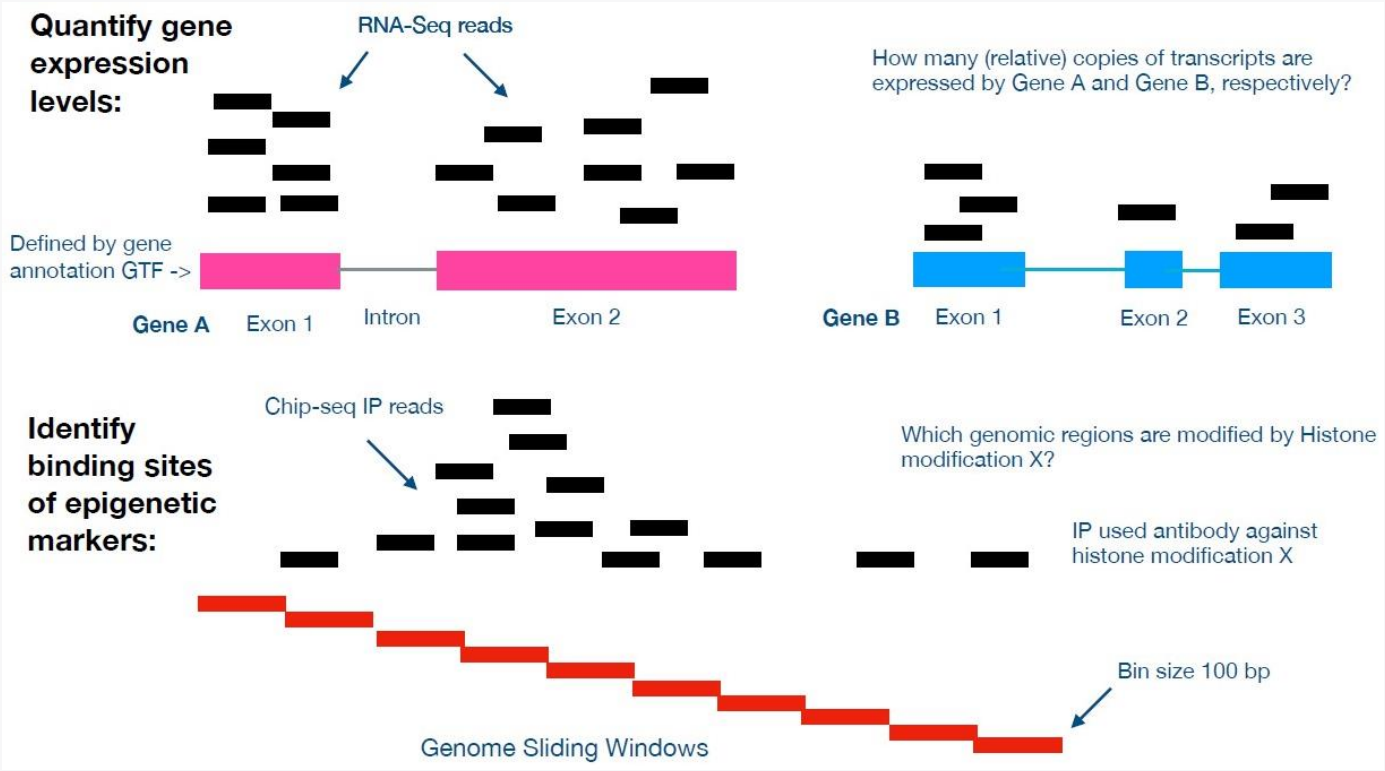
IGV 是一个基于桌面的基因组可视化工具，可以用于展示和分析各种 NGS 数据，包括比对结果、变异和表达数据等。

在 IGV 中，可以通过打开 BAM 文件来显示比对的结果。在展示的 BAM 文件中，每个读都可以被单独选择和查看。可以通过将鼠标悬停在某个读的上方来查看有关该读的详细信息，例如序列和质量值。此外，IGV 还可以生成各种类型的图表来可视化读的覆盖度和分布。

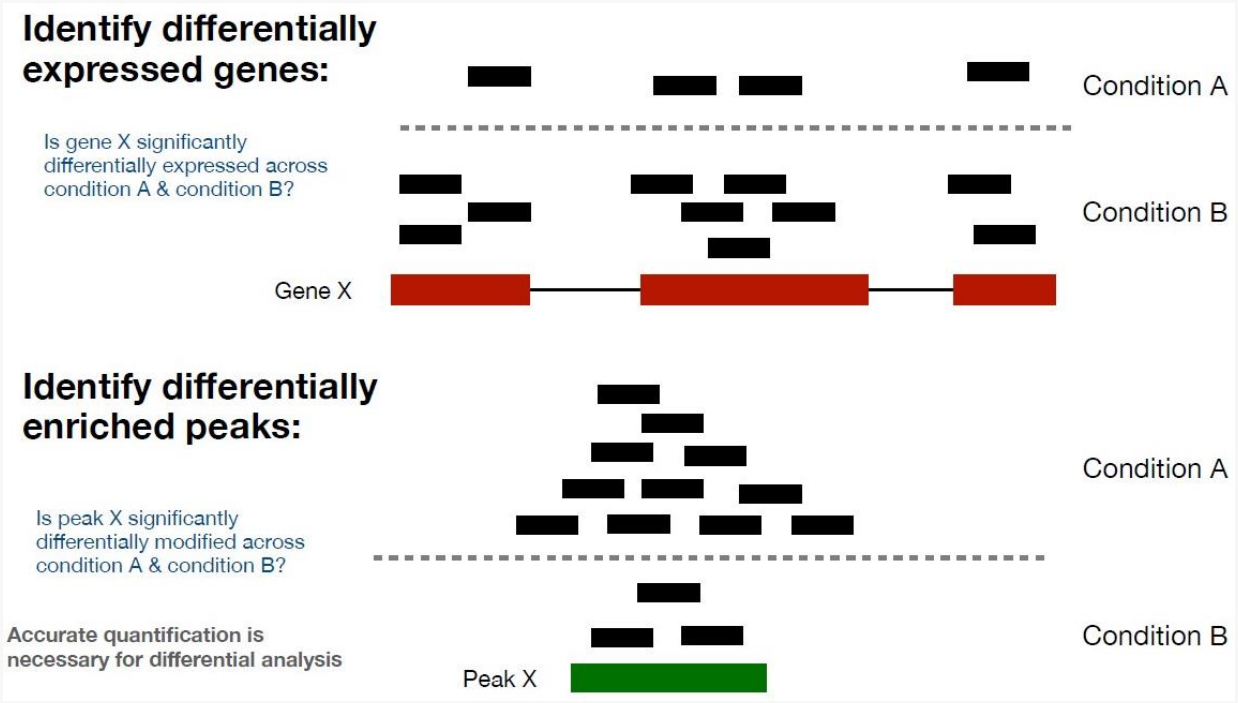
要查看对齐的读数，可以在 IGV 中打开 BAM 文件，然后选择所需的区域。IGV 将在底部显示该区域的覆盖度图表，该图表显示每个位置的读数。您还可以使用右键单击选项菜单中的"Coverage"选项来打开一个新的窗口，该窗口将显示覆盖度图表的更详细视图，其中包括每个读的位置和方向信息。



After genome mapping biological questions can be answered by:



Genome Sliding Windows 是一种计算基因组序列上各种特征的方法，它将基因组序列划分为一系列大小相等的窗口，并对每个窗口中的序列进行特征计算。通常，这些特征计算包括 GC 含量、序列复杂度、单核苷酸多态性等等。这些计算可以用于比较不同基因组区域之间的特征差异，也可以用于在同一基因组区域内寻找特征变化。



Genome mapping 是指将 DNA 序列信息映射到一个参考基因组上的过程。在这个过程中，通过对样品的测序数据进行比对和分析，确定每个 DNA 序列片段在参考基因组上的位置和方向，从而对样品进行基因组注释和分析。这个过程通常包括了测序数据质量控制、比对、去重、排序和格式转换等步骤，可以应用于不同类型的测序数据，如 DNA 测序、RNA 测序、甲基化测序等，以及不同物种的基因组研究。通过 genome mapping，研究者可以了解样品的基因组结构和特征，发现基因、外显子、内含子、调控元件等基因组元素，分析基因组变异和表达等生物学问题。

基于高通量测序（HTS）数据进行的生物信息学分析流程:

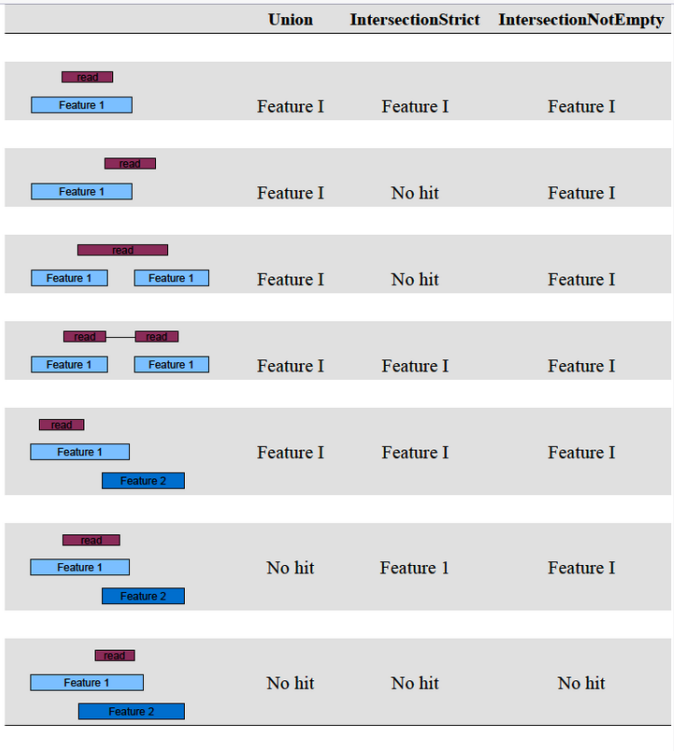
1. 对测序数据进行质控和预处理，包括去除低质量序列、去除接头序列、剪切序列等。此步骤可使用软件如 FastQC 和 Trim Galore。
2. 将预处理后的测序数据进行比对，通常使用软件如 Bowtie2 和 BWA-MEM 等将测序数据比对到参考基因组上，生成 SAM/BAM 格式的比对结果。
3. 对比对结果进行排序和去重，通常使用软件如 SAMtools 和 Picard 等进行处理。
4. 对去重后的比对结果进行基因表达量定量，通常使用软件如 featureCounts 和 HTSeq 等统计每个基因的 reads 计数。
5. 对表达量数据进行归一化和差异表达分析，通常使用软件如 DESeq2 和 edgeR 等进行差异表达分析，得到差异表达基因列表。
6. 对比对结果进行进一步的分析，如基因组范围内的调控元件的寻找，可以使用 ChIP-seq 或 ATAC-seq 等方法进行实验，并使用 MACS2 或 PeakSeq 等软件识别出绑定位点。然后可以使用软件如 ChIPseeker 和 HOMER 等进行调控元件和基因的关联分析，最终得到差异富集的区域（peaks）和调控元件（enhancers，promoters 等）的列表。
7. 将差异表达基因和差异富集的区域进行功能注释，通常使用软件如 DAVID 和 Enrichr 等进行 GO 富集和 KEGG 通路分析，得到基因和区域在生物学过程中的功能注释。

HTSeq Count & R summarizeOverlaps

在 HTSeq Count 或者 R summarizeOverlaps 中，可以选择以下主要的 3+1 种模式：

1. Union mode: 以所有重叠的区域的并集为基础，将 read 分配给所有重叠的基因。这种模式可以用于处理重叠转录本的情况，同时可能导致 reads 被分配给多个基因。
2. Intersection mode: 只将 read 分配给与它们完全重叠的基因，即只考虑在基因内的 reads。这种模式可能会忽略一些潜在的重要信息，如 reads 覆盖的区域比基因的区域宽。
3. IntersectionNotEmpty 则表示在 R 中进行计数时，只有两个区间存在交集（即它们不是完全独立的）时才会进行计数。这个选项通常用于避免多个基因共享同一条 reads 时的计数误差。在 HTSeq Count 中，IntersectionNotEmpty 模式可以通过设置参数 `mode="intersection-nonempty"` 来实现。

4. Same-stranded mode: 只计算与基因在同一条链上的 reads，将另一条链上的 reads 排除在外。这种模式可以用于分析一些组成高度异构的样本，如单个细胞测序数据。

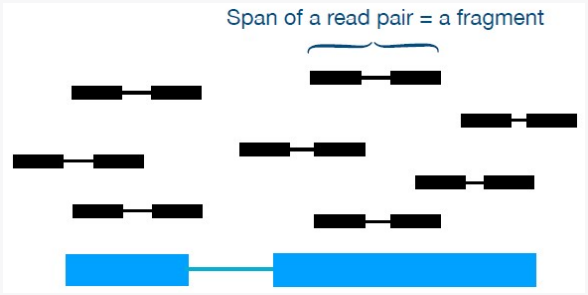


	Union	IntersectionStrict	IntersectionNotEmpty
read / Feature 1	Feature I	Feature I	Feature I
read / Feature 1	Feature I	No hit	Feature I
read / Feature 1 Feature 1	Feature I	No hit	Feature I
read read / Feature 1 Feature 1	Feature I	Feature I	Feature I
read / Feature 1 Feature 2	Feature I	Feature I	Feature I
read / Feature 1 Feature 2	No hit	Feature 1	Feature I
read / Feature 1 Feature 2	No hit	No hit	No hit

Fragment count & read count

Fragment count 和 read count 都是基于测序数据的计数方法，常用于 RNA-seq 和 ChIP-seq 等实验中。它们的计算方式略有不同，主要区别在于是否考虑测序 reads 的 pairing 信息。

在双端测序的实验中，一个 DNA 片段会被分为两个短的 reads 进行测序，这两个 reads 分别称为 mate1 和 mate2。在 Fragment count 的计数方法中，同一 DNA 片段的 mate1 和 mate2 会被计算为一个 fragment，因此只会算作一个计数。而在 read count 的计数方法中，mate1 和 mate2 被看作两个独立的 reads，因此会被分别计算，即算作两个计数。



举个例子，假设有一个双端测序的 RNA-seq 样本，测序得到了如下 reads：

mate1: AAAAAGTTGCTGATCTACCC　　mate2: GGTTGAGTCCTAGCTCTAAA

在 Fragment count 的计数方法中，这两个 reads 来自同一个 RNA 分子，因此只会算作一个 fragment count。而在 read count 的计数方法中，这两个 reads 被看作两个独立的 reads，因此会被分别计算，即算作两个 read count。

总的来说，Fragment count 和 read count 都可以用来评估基因或区域的表达或富集程度，但是在分析时需要根据实验的特点和需求选择合适的计数方法。在双端测序的实验中，Fragment count 方法可以更准确地反映出 RNA 或 DNA 片段的数量，因此在 RNA-seq 和 ChIP-seq 数据分析中比较常用。

EM algorithm

EM 算法（Expectation-Maximization algorithm）是一种迭代求解极大似然估计或最大后验概率的算法。EM 算法的核心思想是通过给定的观测数据，估计模型的参数。但是当模型包含隐变量（即观测不到的变量）时，这个问题变得更加复杂。EM 算法通过迭代估计隐变量，并更新参数，直到收敛到一个局部最优解。

EM 算法包括两个步骤：E 步和 M 步。E 步计算在当前参数下每个隐变量的条件概率分布，即计算观测数据对隐变量的概率分布。M 步则是根据 E 步计算出来的概率分布，来更新参数估计值。

用于生物领域，EM 算法首先需要确定同一基因的不同转录本的长度及其相对丰度，然后根据测序数据，计算每个转录本的覆盖度（coverage）和覆盖范围（span）。根据覆盖度和覆盖范围的比值可以得到该转录本的长度估计值。接着，利用这些长度估计值，可以将所有转录本的覆盖范围在基因组上分段，并统计每个转录本在每个基因组段中的读数。这些读数被用来估计每个转录本的表达量。在计算过程中，EM 算法可以同时考虑不同转录本的长度和表达量，并根据测序数据反复迭代更新这些参数，直到收敛为止。

具体地，EM 算法的步骤如下：

1. 初始化模型参数。
2. E 步：根据当前参数估计值，计算隐变量的条件概率分布。
3. M 步：根据 E 步计算出来的隐变量的条件概率分布，更新模型参数的估计值。
4. 重复执行 E 步和 M 步，直到收敛或达到迭代次数。

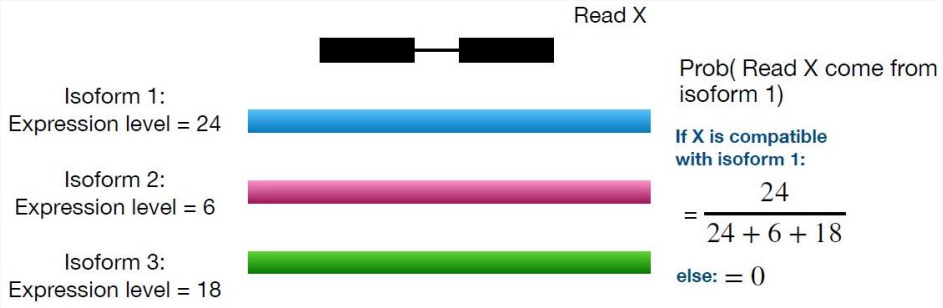
	Tx isoform 1	Tx isoform 2	Tx isoform 3
Read 1	1 0.8	0 0	1 0.2
Read 2	0 0	1 0.6	1 0.4
Read 3	0 0	0 0	1 1
Read 4	1 0.7	0 0	1 0.3
Estimated abundance	0.8+0+0+0.7	0+0.6+0+0	0.2+0.4+1+0.3

Red number: Probabilities reads coming from each transcript

在 EM 算法中，E 步（Expectation step）的目标是计算每个样本在每个转录本中的表达量分布，即计算每个样本在每个转录本上的期望表达量。

以同构体水平量化为例，对于每个样本和每个基因，假设有k个同源的转录本，可以定义一个k维的向量$\mathbf{p}$，其中每个分量p_i表示第i个转录本在该基因下的表达量占比。那么，在 E 步中，需要计算每个样本在每个转录本下的期望表达量，也就是对于每个样本和每个基因，需要计算一个k维的向量$\mathbf{q}$，其中每个分量q_i表示第i个转录本在该样本下的期望表达量。

具体计算方法为：对于每个样本和每个基因，根据当前模型的参数和每个转录本的表达量占比，计算出每个转录本在该样本下的期望表达量，然后将这些期望表达量归一化得到q_i。



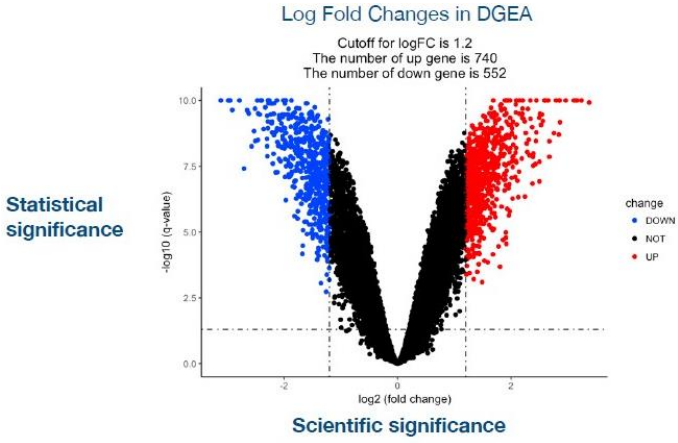
$$q_i = \frac{p_i \times F_i}{\sum_{j=1}^{k} p_j \times F_j}$$

其中，F_i表示第i个转录本的长度归一化后的 reads 计数，p_i表示第i个转录本在该基因下的表达量占比，k为该基因的同源转录本数。在 E 步计算出每个样本和每个转录本的期望表达量后，即可进入 M 步更新模型的参数。

EM 算法的优点在于可以在包含隐变量的模型中求解最大似然估计或最大后验概率估计，并且不需要对隐变量进行直接观测。它的缺点在于需要进行多次迭代，收敛速度较慢，而且容易陷入局部最优解。

Ratio based quantities

"Ratio based quantities" 是基于比值的量化方法，用于比较两个不同样本之间的基因表达水平。在这种方法中，对于每个基因，在两个样本中的基因表达量会被测量，并计算它们之间的比值。这个比值通常被称为差异倍数（fold change），用来表示两个样本之间的表达水平差异。这种方法通常适用于小规模的数据集和表达差异明显的基因，但是不适用于大规模的数据集和表达水平相似的基因。



Log Odds Ratio（LogOR）通常用于二分类问题，例如对照组和实验组的比较。它表示两个组之间某个基因的相对表达量的对数比值，公式为:

$$LogOR = log_2\left(\frac{\frac{n_1}{N_1}}{\frac{n_2}{N_2}}\right)$$ $$LogOR = log_2\left(\frac{\frac{n_1}{N_1}}{\frac{n_2}{N_2}}\right)$$

其中，n_1和N_1表示对照组中该基因的读数和总读数，n_2和N_2表示实验组中该基因的读数和总读数。如果 LogOR 的值为正，则表示该基因在实验组中表达水平更高，否则表示在对照组中表达水平更高。

Log Fold Changes（LogFC）通常用于多组之间的比较。它表示两个组之间某个基因的表达量变化的对数比值，公式为：

$$LogFC = log_2\left(\frac{\frac{n_2}{N_2}}{\frac{n_1}{N_1}}\right)$$ $$LogFC = log_2\left(\frac{\frac{n_2}{N_2}}{\frac{n_1}{N_1}}\right)$$

其中，n_1和N_1表示第一个组中该基因的读数和总读数，n_2和N_2表示第二个组中该基因的读数和总读数。如果 LogFC 的值为正，则表示该基因在第二个组中表达水平更高，否则表示在第一个组中表达水平更高。

需要注意的是，LogOR 和 LogFC 都是对数值，可以帮助我们更好地解释和比较基因表达量之间的差异。在实际应用中，它们可以用于筛选显著差异表达基因，并进一步进行生物学解释。

M-level

M-level，也称为甲基化水平（methylation level），是用于衡量 DNA 甲基化的一种常见方法。DNA 甲基化是一种生物化学修饰，它涉及在 DNA 分子中添加一个甲基基团。这种修饰通常发生在 CpG 位点（即 DNA 分子中的一个特定的碱基序列）上。

M-level 通常是指在给定的 CpG 位点上，甲基化的 DNA 分子的比例。例如，如果在一个给定的 CpG 位点上有 100 个 DNA 分子，其中有 40 个 DNA 分子被甲基化，则该位点的 M-level 为 40%。M-level 的范围通常从 0%（即所有 DNA 分子都没有被甲基化）到 100%（即所有 DNA 分子都被甲基化）。

在研究中，M-level 是一种常见的方法，用于比较不同样本或条件下的 DNA 甲基化。例如，研究人员可能会比较两组细胞或组织中某些基因的 M-level，以了解它们在不同条件下的表达模式。此外，M-level 也可用于评估 DNA 甲基化的整体模式，例如在肿瘤细胞中的变化。

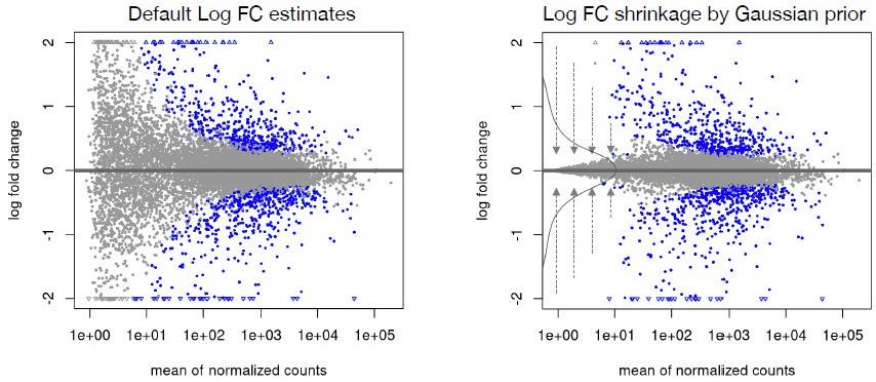
Shrinkage estimator for ratio

在 RNA-seq 数据分析中，差异表达分析是一个重要的任务，它旨在确定不同样本之间的基因表达水平的差异，从而识别与特定条件相关的生物学变化。Shrinkage estimator for ratio 是 RNA-seq 差异表达分析中常用的技术之一，可以帮助研究人员准确地估计基因表达差异。

Shrinkage estimator for ratio 的目标是估计两个样本之间的基因表达比率的差异，该差异被称为基因表达的 fold change。在差异表达分析中，fold change 通常用于评估基因在两个不同条件下的表达变化程度。具体来说，它是两个条件下的基因表达水平的比值，表示一个条件下的表达水平相对于另一个条件下的表达水平的倍数。

在 Shrinkage estimator for ratio 中，首先需要对两个样本的基因表达数据进行预处理，包括读取、过滤、归一化和转换。然后，计算每个基因的表达量比率，并使用这些比率来进行 Shrinkage estimator for ratio 估计。通常使用 log2 比率来进行估计，因为它可以简化估计和比较过程，并且在大多数情况下可以提高结果的准确性。

Shrinkage estimator for ratio 的核心思想是将所有基因的 fold change 估计值“收缩”到一个中心位置，同时保留每个基因的个体差异。这种“收缩”可以减少估计误差，并提高结果的准确性。收缩的大小由先验分布中的方差决定，方差越大，则“收缩”效应越强，而方差越小，则估计值与点估计之间的差异就越小。



在 Shrinkage estimator for ratio 中，先验分布通常选择为某种均匀或正态分布，并通常需要通过实验或先前研究来确定先验分布的方差。一般来说，较小的方差会导致更强的“收缩”效应，从而减少差异估计值之间的差异，提高结果的准确性。然而，如果先验分布的方差设置过小，可能会导致一些真实的差异基因被错误地标记为非差异基因，因此需要进行平衡。

Shrinkage estimator for ratio 通常与其他差异表达分析技术结合使用，如假设检验和多重比较校正等。使用 Shrinkage estimator for ratio 估计值来进行假设检验可以提高分析结果的稳定性和准确性。在多重比较校正方面，Shrinkage estimator for ratio 估计值可以帮助减少假阳性结果的数量，从而提高分析的可靠性。

总之，Shrinkage estimator for ratio 是一种在 RNA-seq 差异表达分析中常用的技术，可以帮助研究人员准确地估计基因表达差异。它的核心思想是通过“收缩”所有基因的 fold change 估计值来减少估计误差，并提高结果的准确性。