四、统计表达值

考虑到小 RNA 测序得到的 exRNA 数据具有碎片化和不均匀的特点，exSeek 用将算法 **??**应用于小 RNA 测序数据，从而构建表达矩阵。对于长 RNA 测序数据，exSeek 用经典的 featurecount 算法来构建表达矩阵。参数设置为"`-t exon -g gene_id -M -s`"。

五、表达矩阵处理

1. 过滤低表达特征

由于受限于二代测序技术测序深度只有部分基因被检测到的特点，再加上只有部分基因被表达的原因，exSeek 分析长 RNA 测序数据得到的表达矩阵有很多特征（通常为基因）的表达值是零。这些特征不仅干扰后续的分析，还增加分析所需的资源。所以，exSeek 过滤了低表达的特征。exSeek 可以自定义地选择过滤参数，默认设置为过滤掉表达量小于 5 占总样本数超过 50% 的特征。exSeek 还可以设置过滤模式，输入是表达量为序列数目的表达矩阵，可以经过基于序列数目的过滤，也可以是基于 counts per million（CPM）的过滤，或是基于 reads per kb per million reads（RPKM）的过滤，最后返回表达量为序列数目的表达矩阵。

2. 零值插补

类似于单细胞测序，由于测序 RNA 的量很小，exRNA-seq 数据也会出现过量的零，即缺失值（dropout）。而下游的种种分析比如差异表达分析，都要求基因表达测量的准确性。因此，修正实则有表达而表现为零表达的情况很重要。exSeek 选取了两种方法来插补零值：

- scImpute
- VIPER

（1）scImpute

scImpute 的核心思路是：

- 确定每个样本每个基因的表达是否为缺失值；
- 借用同一基因相似样本的表达值信息来填补缺失值。

对于样本 j，它有高可能性的表达基因集 B_j，也有表达值为缺失值的基因集 A_j，N_j 为包含样本 j 的距离较近的样本集合，则权重计算如下：

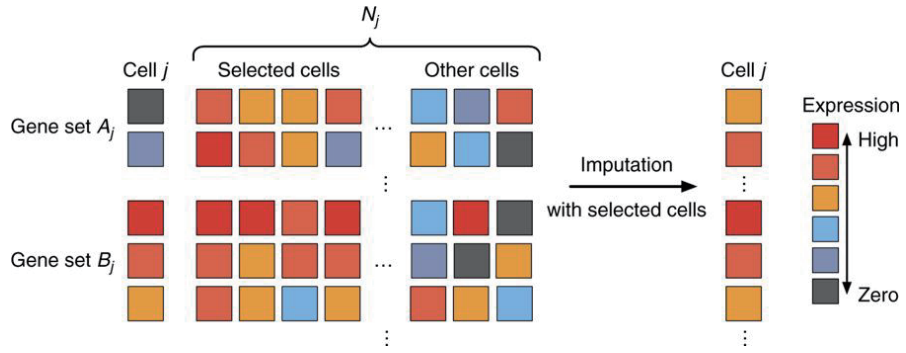

图 2.2　scImpute 原理图[?]

注：通过拟合混合模型，scImpute 首先估计每个样本每个基因表达缺失的可能性。为了找到填补基因表达的相似样本，要保障样本之间距离不受缺失值影响，scImpute 对表达矩阵进行主成分分析（principal components anlysis, PCA），PCA 选择解释性比较大的几个主成分，这些主成分是基因的线性组合，它们被用来后续对样本的聚类。对于每个样本，有些基因具有高可能性的表达值，有些基因相反，scImpute 计算距离较近的样本的权重，对高可能性基因做回归，随后将权重作用于低表达基因上，缺失值得到了插补。

$$\hat{\boldsymbol{\beta}}^{(j)} = \mathrm{argmin}_{\beta^{(j)}} \left\| \mathbf{X}_{B_j,j} - \mathbf{X}_{B_j,N_j}\boldsymbol{\beta}^{(j)} \right\|_2^2, \text{ subject to } \boldsymbol{\beta}^{(j)} \geqslant \mathbf{0} \tag{2.2}$$

计算出来的权重作用于低表达基因：

$$\hat{\mathbf{X}}_{ij} = \mathbf{X}_{i,N_j}\hat{\boldsymbol{\beta}}^{(j)}, \quad i \in A_j \tag{2.3}$$

这就完成了表达缺失基因的插补。

（2）VIPER

VIPER[?] 的思路与 scImpute 十分相似，但存有以下几点区别：

3. 库大小归一化

解决表达矩阵的零表达问题以及缺失值问题后，其还存有样本间的差异。差异来源之一是库大小（library size）不均一，即测序深度不一使得下机序列数量没有可比性。为了解决这个问题，exSeek 采用了 5 种方法来规范化表达矩阵：

- CPM
- CPM_{top}
- UQ
- TMM
- RLE

其中，CPM 和 CPM_{top} 是用 R 实现；UQ，TMM 和 RLE 用 edgeR 实现[?]。

表 2.2　scImpute 与 VIPER 的比较

区别	scImpute	VIPER
混合模型	Γ 分布和正态分布的混合，混合权重与缺失可能性有关	零膨胀泊松分布和零值的混合，混合权重与缺失可能性有关
选择相似样本	利用 PCA 降维，再聚类出距离相近的样本	根据插补信息来源的权重，设立阈值，若过小，则权重为 0，意味着这些样本对插补没有影响，不具对称性
插补信息作用形式	权重 > 0	权重 $\in (0,1)$，权重之和等于 1

注：对称性指的是，计算样本众多缺失表达基因时，会有距离较近的样本作为插补信息来源，反过来，对于这些后者，考虑来源时，并不能保证前者是插补信息来源。

为了后续表述方便，数量（counts，对应有 RPM 等等）表达矩阵记为：

$$\mathbf{C}_{g\times s}$$

，其中矩阵元素 $c_{ij}, i \in [1,g], j \in [1,s]$ 其中，g 是表达矩阵基因数量，s 是表达矩阵样本数量。

（1）CPM

CPM 用的是 counts per million（CPM）方法。该方法计算每个样本表达量之和，再乘以一个比例因子，使得表达量之和为1×10^6。这种对每个样本乘以比例因子的方法是最简单的归一化方法。可以写成：

$$\mathbf{C}^*_{g\times s} = \mathbf{C}_{g\times s} \times \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_s \end{bmatrix} \text{ subject to } \sum_{i=1}^{g} c^*_{ij} = 1\times10^6 \tag{2.4}$$

（2）CPM_{top}

类似于 CPM，CPM_{top} 也是每个样本下乘以一个比例因子。只不过 CPM 作用之后，每个样本的表达量之和为1×10^6，而 CPM_{top} 将基因分成高表达低表达两部分，两部分乘的比例因子不同，使得每个样本高表达和低表达的基因表达量之和都为1×10^6，即

$$\mathbf{C}_{g_{\text{high}}\times s} \times \begin{bmatrix} \lambda_1^{\text{high}} & & \\ & \ddots & \\ & & \lambda_s^{\text{high}} \end{bmatrix} \text{ subject to } \sum_{i\in\{\text{highly-expressed genes}\}} c^*_{ij} = 1\times10^6 \tag{2.5}$$

$$\mathbf{C}_{g_{\text{low}}\times s}\times\begin{bmatrix}\lambda_1^{\text{low}} & & \\ & \ddots & \\ & & \lambda_s^{\text{low}}\end{bmatrix}\text{ subject to }\sum_{i\in\{\text{low-expressed genes}\}}c_{ij}^{*}=1\times10^{6}\tag{2.6}$$

CPM_{top} 进行这样的操作是因为：exRNA 测序数据中前 20 个高表达的基因占据总表达量的大约 50%，然而很多有意义的基因都不是最高表达的基因，它们的归一化受高表达基因影响很大，CPM_{top} 理论上可以降低影响。

（3）UQ

CPM 带来的问题是，归一化的比例因子受不关心的成分（高表达基因或者不感兴趣的基因例如污染基因）的影响：如果这些成分存在，那么归一化的比例因子就会受到平均。如果这些成分高表达并且差异表达，那么比例因子主要作用对象是它们而不是感兴趣的基因。UQ 采用 upper quartile 的方法，首先去除所有样本中表达均为 0 的基因，再从剩下的基因逐样本选出上四分位点作为该样本库的大小归一化[?]。这样，比例因子不会受特别高表达基因的影响。(UQ)

（4）TMM

TMM，trimmed mean of M-values，力图解决测序污染（混有无用的下机序列）等情况导致的前述比例因子作用不大即无法将真实有用的基因数目归一化的情况。TMM 假设两个样本中（j_1, j_2）大部分基因没有差异表达，则 $\boldsymbol{M}_i$ 系数：

$$\boldsymbol{M}_i=\log_2\frac{c_{ij_1}/\sum_{i=1}^{g}c_{ij_1}}{c_{ij_2}/\sum_{i=1}^{g}c_{ij_2}}\tag{2.7}$$

大部分都为 1。去除高表达的基因，再去掉两样本间差异比较大的基因（M 值较大），对留下来的 $\boldsymbol{M}_i$ 值做加权平均，便得到 TMM 值。该值可作为两样本间的校正系数[?]。在测序污染情况下或者高表达差异表达情况下，被丢掉的 $\boldsymbol{M}_i$ 值正好对应不需要的基因，因此标准化不会有很大的偏移。

（5）RLE

RLE[?] 关心的问题与 TMM 等关心的问题一致，都是要解决 CPM 没有考虑到的问题：表达量高且差异表达的基因会对归一化过程产生干扰。与 TMM 一致，RLE 的假设也是样本间差异表达的基因只占少数。每个样本的库大小计算如下：对于第 j 个样本，基因表达值为分子，该基因下所有样本表达值的几何平均作为分母，此表达式以基因为变量，对基因取中位数，则为该样本的库大小再用于归一化。

$$\hat{s}_j=\text{median}\frac{c_{ij}}{\left(\prod_{v=1}^{s}c_{iv}\right)^{1/s}}\tag{2.8}$$

其中，$\hat{s}_j$ 为样本 j 的库大小。

4. 批次效应去除

库大小归一化解决的是测序深度不同带来的问题，其无法解决不同文库制备方法带来的差异以及其他产生干扰的复杂效应的影响。当数据的来自很复杂的实验时，比如不同的细胞分选时间，不同测序批次以及不同测序中心等等，表达矩阵不仅仅需要考虑测序深度不同带来的问题，也要校正这些批次效应带来的影响。exSeek 采用了三种方法：

- limma
- ComBat
- RUV

（1）limma

limma 提供了 `removeBatchEffect` 函数用来校正批次效应[?]。limma 基于线性回归，认为每个基因的表达值包含两个成分，真实存在的表达值和样本的批次效应附加的表达值。limma 以批次信息作为输入，设计了一个包含批次效应对应的设计矩阵（design matrix，又称为 blocking factor）的线性模型。用每个基因的对数表达值拟合（某批次效应下的所有表达值更贴近正态分布），随后，将线性模型设计矩阵项的系数设为 0，从余项计算表达值，产生的矩阵就是去除批次效应的表达矩阵。

（2）ComBat

与 limma 类似，ComBat 也是基于线性回归的办法[?]。但 ComBat 执行额外的操作：它采用经验贝叶斯方法（empirical Bayes），收缩了线性模型中批次效应项的系数。ComBat 认为标准化后的矩阵相同批次的批次效应，众多基因的该批次效应值服从正态分布，利用这样的先验信息，可以批次效应生成后验作为校正后的批次效应值，然后表达值减去校正后的批次效应值就完成了批次效应校正的操作。这么做的好处是：对每个基因做批次效应的估计时，ComBat 从别的基因的批次效应值借信息。如果该基因批次效应值偏差很大，由于借来的信息的中和作用（先验对后验的作用，先验包含了其他基因的信息，似然是该基因的信息），不会产生批次效应的过校正的现象，这就起到了收缩批次效应项系数的作用。

（3）RUV

在不知道批次效应信息时，ComBat 和 limma 无法通过线性回归来进行批次效应的校正。RUV（remove unwanted variation）可以在未知变异因素情况下校

正批次效应。RUV 采用的不再是线性拟合的方法而是因子分析（factor analysis）的方法。RUV 认为，表达值由两部分组成，一部分是感兴趣变异因素贡献的，另一部分由要去除的变异因素贡献。组合的权重以这些变异因素为变量。记为：

$$\log E[\mathbf{C}^{\top}|\mathbf{W},\mathbf{X}] = \mathbf{W}\boldsymbol{\alpha} + \mathbf{X}\boldsymbol{\beta} \tag{2.9}$$

$\mathbf{C}$ 是 $g \times s$ 大小的表达矩阵，$\mathbf{X}$ 是 $s \times p$ 大小的矩阵，其中 p 对应感兴趣的变异因素比如样本处理方法的数量，$\boldsymbol{\beta}$ 是 $p \times g$ 大小的参数矩阵，$\mathbf{W}$ 是 $s \times k$ 大小的矩阵，其中 k 对应需要去除的变异因素的数量，$\boldsymbol{\alpha}$ 是 $k \times g$ 大小的参数矩阵。

RUV 包含 RUV_g，RUV_r 以及 RUV_s。它们具体的应用不同：

RUV_g

RUV_g 可以在已知负对照基因比如非差异表达基因情况下进行批次效应的校正[?]：用奇异值分解（singular value decomposition, SVD）后取最大 k 个奇异值（singular values）对应的矩形对角矩阵的办法从对照基因的数据估计设计矩阵 $\mathbf{W}$，$\mathbf{W}$ 服从 $\log E[\mathbf{C}_{\text{ctr}}^{\top}|\mathbf{W},\mathbf{X}] = \mathbf{W}\boldsymbol{\alpha}_{\text{ctr}}$，然后 $\mathbf{W}$ 回代方程 **??**，用广义线性模型（generalized linear model, GLM）计算出 $\mathbf{W}$ 的参数 $\boldsymbol{\alpha}$，这样就完成了批次效应的估计。

RUV_r

RUV_r 提供一种不依赖于负对照基因计算要去除的因素的设计矩阵的方法：用 GLM 对表达值 $\mathbf{C}^{\top}$ 做感兴趣因素设计矩阵 $\mathbf{X}$ 的回归，

$$\log E[\mathbf{C}^{\top}|\mathbf{W},\mathbf{X}] = \mathbf{X}\boldsymbol{\beta} + \text{deviance residuals} \tag{2.10}$$

残差项包含了要去除因素的成分。对残差矩阵做 SVD 分解，取最大 k 个奇异值（singular values）对应的矩形对角矩阵的办法来估计要去除的因素的设计矩阵 $\mathbf{W}$，之后的操作与 RUV_g 同。

RUV_s

RUV_r 提供一种不依赖于负对照基因却依赖负对照样本（重复样本）的方法计算要去除的因素的设计矩阵：重复样本组内的表达值进行中心（center）化，对这种中心化的表达矩阵做 SVD，最大 k 个奇异值（singular values）对应的矩形对角矩阵从对照样本的信息中估计矩阵 $\mathbf{W}$ 的参数 $\boldsymbol{\alpha}$，用最小二乘法（ordinary least squares，OLS）对选定的负对照基因做估计得到设计矩阵 $\mathbf{W}$，之后的操作与 RUV_g 同。这个方法更加稳健，因为其需要两种负对照的信息。

5. 矩阵处理效果评估

对矩阵处理最初的目的：

- 去除不感兴趣变异因素（批次效应，测序深度）
- 保留感兴趣变异因素（不同实验处理下的生物学差异）

exSeek 提出两个指标 mkNN 和 UCA 来评估表达矩阵是否达到了我们的目的。

（1）mkNN

mkNN 是用来评估批次效应大小的指标。选取一个样本，如果它周围的样本都从属同一批次，对于多数样本都是如此，则表明批次效应显著。在只有两个批次情况下，已有研究根据此想法提出 Alignment Score[?]，

$$\text{Alignment Score} = \frac{1}{k - \frac{k}{N}}(k - \overline{x}) \tag{2.11}$$

其中 k 是最近邻算法（k nearest-neighbors, kNN）的前 k 个最近邻，$\overline{x}$ 是样本周围的样本同属一个批次的数量的平均，N 表示样本数。当两个批次样本完全分开时，$k = \overline{x}$，Alignment Score $= 0$；当两个批次样本完全混杂时，比例因子 $\frac{1}{k-\frac{k}{N}}$ 作用下，Alignment Score 接近 1。exSeek 提出了适用于多种批次的 mkNN 指标①。

$$\text{mkNN} = 1 - \frac{1}{B}\sum_{b=1}^{B}\frac{\overline{x}_b - kN_b/(N-1)}{\min\left(k, N_b\right) - kN_b/(N-1)} \tag{2.12}$$

其中，b 表示批次，B 为批次数量，N_b 是批次 b 下样本的数量。批次效应越明显，该指标越接近 0。

（2）UCA

exSeek 采用评估无监督学习指标 UCA（unsupervised clustering accuracy）来检测聚类分得的类与真实类别之间的差异。矩阵通过降维，再经 K-means 算法进行聚类，聚类分出的类别即预测的类别，如果和真实类别匹配上，则表达矩阵的生物学差异可以被这种无监督学习检测到。然后，聚类得到的类别是不加属性标签的，真实类别要如何匹配到聚类得到的类别，这个问题是算法中的二分图的最大匹配问题。匈牙利算法（Hungarian algorithm）可以解决这个问题。匹配之后，建立混淆矩阵（confusion matrix），计算准确度。

六、机器学习

经过以上库大小归一化以及批次效应的去除后，表达矩阵可以作为这步的输入，经过机器学习，挑选出标志物，并且对标志物有相应的评价。机器学习包括②：

①该指标由由史斌斌（ltbyshi@gmail.com）首次提出

②这部分工作由史斌斌（ltbyshi@gmail.com）首次探索