Chapter 7. Normalization

7.1 Motivation

* Systematic differences in sequencing coverage between libraries
* Arise from technical differences in cDNA capture or PCR amplification efficiency across cells
  + PCR: Polymerase chain reaction (PCR) is a method widely used in molecular biology to rapidly make millions to billions of copies of a specific DNA sample allowing scientists to take a very small sample of DNA and amplify it to a large enough amount to study in detail.
* Difference between normalization and batch correction
  + Normalization occurs regardless of the batch structure and only considers technical biases
  + Batch correction only occurs across batches and must consider both technical biases and biological differences
* Scaling normalization
  + dividing all counts for each cell by a cell-specific scaling factor, often called a “size factor”
  + Assumption
    - Any cell-specific bias affects all genes equally via scaling of the expected mean count for that cell.

7.2 Library size normalization

* Library size
  + The total sum of counts across all genes for each cell, the expected value of which is assumed to scale with any cell-specific biases.
  + librarySizeFactors
    - Proportionality constant is defined such that the mean size factor across all cells is equal to 1
  + The use of library size factors assumes that there is no “imbalance” in the differentially expressed (DE) genes between any pair of cells.
    - any upregulation for a subset of genes is cancelled out by the same magnitude of downregulation in a different subset of genes.
    - DE: A gene is declared differentially expressed if an observed difference or change in read counts or expression levels between two experimental conditions is statistically significant.
    - Not always balanced DE, thus library size normalization may not yield accurate normalized expression values for downstream analyses.

7.3 Normalization by deconvolution

* Two cells where a single gene X is upregulated in one cell A compared to the other cell B
  + more sequencing resources are devoted to X in A, thus decreasing coverage of all other non-DE genes when the total library size of each cell is experimentally fixed
  + the library size of A increases when X is assigned more reads or UMIs, increasing the library size factor and yielding smaller normalized expression values for all non-DE genes.
* estimateSizeFactorsFromMatrix() or calcNormFactors()

7.4 Normalization by spike-ins

* Assumption
  + the same amount of spike-in RNA was added to each cell
* An RNA spike-in is an RNA transcript of known sequence and quantity used to calibrate measurements in RNA hybridization assays, such as DNA microarray experiments, RT-qPCR, and RNA-Seq.
  + A spike-in is designed to bind to a DNA molecule with a matching sequence, known as a control probe.
* Equalize spike-in coverage across cells by scaling with “spike-in size factors”
* Spike-in normalization should be used
  + if differences in the total RNA content of individual cells are of interest and must be preserved in downstream analyses
* computeSpikeFactors()
  + Estimate spike-in size factors for all cells
  + Converting the total spike-in count per cell into a size factor
* If normalized spike-in data is required, we must compute a separate set of size factors for the spike-in transcripts.
  + Automatically performed by functions such as modelGeneVarWithSpikes()

7.5 Applying the size factors

7.5.1 Scaling and log-transformation

* logNormCounts()
  + to compute normalized expression values for each cell
  + Creates “logcounts”, log-values will be the basis of our downstream analyses
* When log-transforming, we typically add a pseudo-count to avoid undefined values at zero.

7.5.2 Downsampling and log-transforming

* Problem: differences in the sizes of the counts
  + downsample the counts of the high-coverage cells to match those of low-coverage cells
  + uses the size factors to determine the amount of downsampling for each cell required to reach the 1st percentile of size factors
  + Downsampling is statistically inefficient
    - it needs to increase the noise of high-coverage cells in order to avoid differences with low-coverage cells
    - Slower than simple scaling

Chapter 8. Feature Selection

8.1 Motivation

* The choice of genes to use in this calculation has a major impact on the behavior of the metric and the performance of downstream methods.
* We want to select genes that contain useful information about the biology of the system while removing genes that contain random noise.
* Simplest way: select the most variable genes based on their expression across the population

8.2 Quantifying per-gene variation

8.2.1 Variance of the log-counts

* The simplest approach to quantifying per-gene variation is to simply compute the variance of the log-normalized expression values (referred to as “log-counts” for simplicity) for each gene across all cells in the population.
* Feature selection requires modelling of the mean-variance relationship

8.2.2 Coefficient of Variation

* An alternative approach to quantification uses the squared coefficient of variation (CV2) of the normalized expression values prior to log-transformation.

8.2.3 Quantifying Technical Noise

* The use of a trend fitted to endogenous genes assumes that the expression profiles of most genes are dominated by random technical noise.
* In the absence of spike-in data, one can attempt to create a trend by making some distributional assumptions about the noise.
* modelGeneVarByPoisson()

8.2.4 Accounting for blocking factors

8.2.4.1 Fitting block-specific trends

* Data containing multiple batches will often exhibit batch effects
* focus on genes that are highly variable within each batch
  + performing trend fitting and variance decomposition separately for each batch

8.2.4.2 Using a design matrix

* If involving a large number of blocking factors and/or covariates
  + use the “design=” argument to specify a design matrix with uninteresting factors of variation

8.3 Selecting Highly Variable Genes

8.3.1 Overview

* A larger subset will reduce the risk of discarding interesting biological signal by retaining more potentially relevant genes, at the cost of increasing noise from irrelevant genes that might obscure said signal.

8.3.2 Based on the Largest Metrics

* The simplest HVG selection strategy is to take the top X genes with the largest values for the relevant variance metric.
* Disadvantage
  + It turns HVG selection into a competition between genes, whereby a subset of very highly variable genes can push other informative genes out of the top set.
  + The choice of X is fairly arbitrary

8.3.3 Based on significance

* Set a fixed threshold of one of the metrics
  + done with the (adjusted) p-value
  + The p-value for each gene is generated by testing against the null hypothesis that the variance is equal to the trend.

8.3.4 Keeping all genes above the trend

* Aim: remove the obviously uninteresting genes with variances below the trend
  + avoid the need to make any judgement calls regarding what level of variation is interesting enough to retain
* This approach represents one extreme of the bias-variance trade-off where bias is minimized at the cost of maximizing noise.

8.4 Selecting a priori genes of interest

* use pre-defined sets of interesting genes
  + focus on specific aspects of biological heterogeneity that may be masked by other factors when using unsupervised methods for HVG selection.
  + Disadvantage: limit our capacity to detect novel or unexpected aspects of variation

8.5 Putting it all together

* Several options
  + Subset the SingleCellExperiment to only retain our selection of HVGs.
  + Keep the original SingleCellExperiment object and specify the genes to use for downstream functions via an extra argument like “subset.row=”
  + Using the “alternative Experiment” system in the SingleCellExperiment class