Quantile Normalization

Brayan Gutierrez

November 11th, 2025

Gene Expression Normalization Workflow

This analysis performs pre-processing and normalization of RNA-seq expression data for control and late-stage samples in an Age-Related Macular Degeneration (AMD) dataset. The workflow implements both global and class-specific quantile normalization, as well as qsmooth normalization, to ensure comparable expression distributions across samples and disease stages prior to downstream co-expression network analysis.

Data Import and Preparation

All analyses were conducted in `R` using the packages `RColorBrewer`, `MEGENA`, `visNetwork`, `igraph`, `dplyr`, `preprocessCore`, and `qsmooth`. Two input files were used:

- **aak100_cpmdat.csv**: CPM-normalized RNA-seq expression matrix with genes as rows and samples as columns.
- **gene_info.tsv**: Gene annotation table containing functional or descriptive metadata.

The `mgs_level` column, representing disease stage (e.g., MGS1 for controls, MGS4 for late AMD), was extracted as a class label, and all remaining columns were treated as expression values. The data matrix was transposed to have samples as rows and genes as columns, and all entries were coerced to numeric format prior to normalization.

Global Quantile Normalization

A custom helper function `quantile_norm()` was defined to perform quantile normalization using the `normalize.quantiles()` function from the `preprocessCore` package. This step enforces identical empirical distributions across all samples, removing global technical variability such as differences in sequencing depth or library size. The resulting normalized matrix, denoted as `expr_QN_global`, represents globally normalized expression values across the entire dataset.

Class-Specific Quantile Normalization

To preserve within-group biological variation, a second normalization function `quantile_norm_by_class()` was implemented. This function applies quantile normalization separately to each disease class defined by `mgs_level`. For each class:

1. Samples belonging to that class were subset from the expression matrix.
2. Quantile normalization was applied only within that subset.
3. The normalized subsets were recombined into a single matrix.

The resulting matrix, `expr_QN_class`, retains stage-specific structure while removing technical bias within each class.

QSmooth Normalization

To balance global and within-group normalization, qsmooth normalization was performed using the `qsmooth()` and `qsmoothData()` functions from the `qsmooth` package. This method computes a data-driven smoothing parameter that determines the relative contribution of global versus class-level normalization, thereby preserving biologically meaningful between-group differences. The resulting normalized expression matrix was stored as `expr_QS`.

Subsetting by Disease Stage

To examine stage-specific expression patterns, the pipeline was repeated for:

- **Control samples (MGS1)**
- **Late AMD samples (MGS4)**

Each subset underwent the same series of normalization steps—global quantile normalization, class-specific quantile normalization, and qsmooth normalization—yielding standardized datasets suitable for subsequent co-expression network construction and differential module analysis using frameworks such as MEGENA or WGCNA.

Summary

In summary, this workflow:

1. Loads and structures RNA-seq expression data.
2. Applies global, within-group, and adaptive (qsmooth) normalization strategies.
3. Produces normalized matrices that are directly compatible with downstream network-based or differential expression analyses comparing transcriptional organization between control and late-stage AMD conditions.