Predicting Organoid Quality with ML

Project: Organoid-Data-Analysis
Notebook: nb5_quality_ml
Inputs: nb4 outputs (factor means per organoid×cluster; gene loadings)

Purpose

Organoids differ in how faithfully they reproduce human developmental programs. Some are closer to fetal tissue, while others deviate or fail to differentiate correctly. Defining "quality" is therefore essential but difficult: it is not a single measurable property, and no universal standard exists.

This notebook demonstrates how organoid quality can be framed as a **machine learning problem**. We use latent factors derived from multi-omic integration (nb4) as features, and construct a balanced synthetic High/Low label to illustrate the workflow. The goal is not to discover real biological truth, but to show a reproducible method for:

- Training models under strict cross-validation.
- Evaluating predictive performance against baselines.
- Quantifying which factors drive predictions.
- Mapping those factors back to gene programs and pathways.

> Note: The machine learning analysis presented here builds directly on the latent factors generated in Notebook 4 (omic integration). To follow this notebook, make sure to first complete the omic integration analysis and export the factor tables, as these form the input for all ML steps.

Challenges in Predicting Organoid Quality

1. **Lack of a ground truth**
   There is no universally accepted definition of organoid quality. Labels must often be inferred or approximated.
   Example: here we simulate High/Low quality labels from latent factors. In real studies, labels could come from fetal reference similarity, morphology scores, or QC assays.

2. **Small sample sizes**
   Organoid experiments usually yield few replicates, making statistical analysis and ML training unstable.
   Example: this notebook uses only 8 organoid×cluster units.

3. **Overfitting risk**
   High-dimensional features combined with small N can lead to models that fit noise.
   Solution: we apply stratified CV, seed stability checks, and label-shuffle baselines to guard against spurious signal.

4. **Interpretability**
   A model is only useful if predictions can be explained biologically.
   Solution: we use permutation importance aligned to AUROC, then trace influential factors back to gene loadings and pathways from nb4.

Framework for Analysis

The workflow is structured into sequential layers that build from simple to complex:

1. **Dataset construction** — Organize organoid×cluster units with factors F1..Fk and add a synthetic High/Low label.

2. **Baseline evaluation** — A majority-class predictor defines chance-level AUROC/AUPRC/Brier that any real model must beat.

3. **Model training** — Random Forest provides a flexible, robust baseline; XGBoost is tested for comparison.

4. **Feature importance** — Permutation ΔAUROC reveals which latent factors carry predictive signal.

5. **Pathway enrichment** — Factor-specific gene loadings are scored and tested for enrichment to link predictions back to biological programs.

6. **Robustness checks** — Repeat analysis with altered seeds and shuffled labels to confirm stability.

Guardrails are built in: stratified CV preserves class balance; low-N checks prevent over-interpretation; shuffled labels ensure we don't confuse noise with signal.

Steps in This Notebook

1. **Setup** — environment, seed, paths, helper functions.

2. **Load features** — import nb4 factor means; add synthetic High/Low labels.

3. **Baseline** — majority-class model; check for redundant features.

4. **Random Forest** — cross-validation, out-of-fold metrics, save predictions.

5. **Permutation importance** — quantify factor contributions.

6. **Pathway mapping** — compute weighted gene scores; run enrichment.
   6b. **Score-based enrichment** — factor-derived sets tested with permutation.
7. **Robustness** — seed stability and label shuffle.
8. **Exports** — save all artifacts and run summary.
9. **Quick report** — reload key results.
10. **XGBoost** — optional second model.
11. **Model card** — record setup, metrics, artifacts in markdown.
12. **Threshold sweep** — identify best operating point and confusion matrices.

Addressing Organoid Quality — Broader Perspective

Predicting organoid quality is fundamentally a **proxy problem**: quality is not directly observable but must be inferred from signals such as fidelity to fetal programs, cellular composition, or technical QC metrics.

This notebook shows how to:

- Formalize quality as a supervised ML task.
- Compare models against strict baselines.
- Translate factor importance into interpretable biological drivers.

What else can be done:

- Replace the synthetic label with real QC measures (transcriptomic fidelity, morphology scoring, lineage-specific markers).
- Increase the dataset with more organoids to strengthen generalizability.
- Expand features to additional modalities (spatial, epigenomic, imaging).
- Explore richer interpretability tools (SHAP, attention-based models, multimodal KPNN).

Biological Context

The factors identified as important in this notebook are shaped by the synthetic label, but the methodology scales to real biology. In practice, predictive factors may correspond to pathways that distinguish higher- and lower-quality organoids (e.g., neurogenesis programs, stress responses, gliogenesis markers). This makes the approach suitable for guiding protocol optimization and QC in organoid studies.

How This Notebook Helps

- Provides a **tutorial-style template** for predicting organoid quality with ML.
- Embeds **baselines and guardrails** to ensure fair evaluation.
- Links **abstract ML factors back to biology**, making results interpretable.
- Produces a **model card** that documents metrics and artifacts for reproducibility.

Next Steps Beyond This Notebook

- Define real-world **quality labels** using experimental metrics.
- Apply the framework to larger, independent datasets.
- Integrate **multi-modal features** for richer prediction.
- Use the model to test whether interventions (protocol tweaks, drug treatments) improve organoid fidelity.

Key Takeaways

- Organoid quality can be modeled as a supervised ML problem.
- Baselines (Dummy) and robustness checks (label-shuffle, seed stability) are critical to avoid false signal.
- A small subset of latent factors often drive predictions, simplifying interpretation.
- The framework is extensible: once real QC labels exist, this pipeline can be directly applied to measure and improve organoid fidelity.