Single-cell Morphology Quality Control (coSMicQC)

Dave Bunten[1*], Jenna Tomkinson[1*], Vincent Rubinetti[1], Gregory Way[1]

[1]*Department of Biomedical Informatics, University of Colorado Anschutz Medical Campus*

**These authors contributed equally to this work.*

Scan for coSMicQC!

Department of Biomedical Informatics
SCHOOL OF MEDICINE
UNIVERSITY OF COLORADO **ANSCHUTZ MEDICAL CAMPUS**

I. Erroneous outliers and analysis

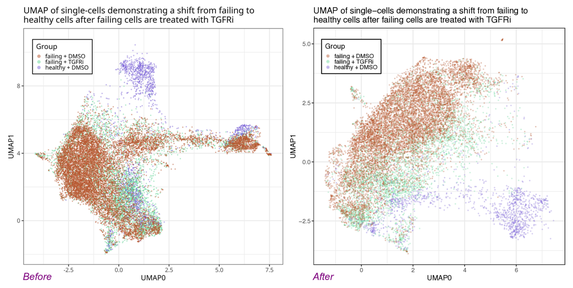
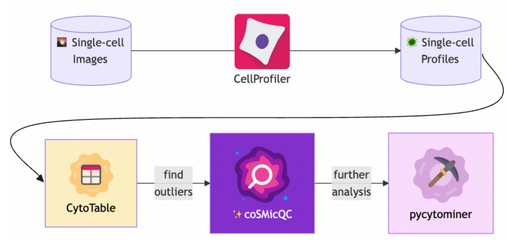


Figure 1: ***Extra clustering islands can be seen when looking at morphological profiles linked to poor segmentation, which when removed, better reveal patterns in the data.***

Segmentation errors during single-cell morphology image analysis such as misidentifying cell compartments or artifacts as cells can lead to inaccurate single-cell measurements and ***erroneous anomalies*** within the data (Figure 1). ***If single-cell quality control is performed, it often uses bespoke methods*** or aggregate data into bulk profiles to avoid discrepancies caused by anomaly outliers. These techniques make it challenging to perform ***quality control*** on the data, impeding the potential for meaningful discoveries.

II. Single-cell quality control package



To address these challenges, we introduce ***coSMicQC (Single-cell Morphology Quality Control)***, an open source Python package designed to enhance the accuracy of single-cell morphology analysis. **coSMicQC** offers default and customizable thresholds for quality control, integrating seamlessly into both command line and Python API workflows.

III. Getting started with coSMicQC

☆ 1) *Installation*

```
# pip install from pypi
pip install coSMicQC

# or install directly from source
pip install git+https://github.com/WayScience/
coSMicQC.git
```

coSMicQC may be installed from PyPI or source.

☆ 2) *Finding outliers*

```
import cosmicqc
# find outliers from single-cell profiles
scdf = cosmicqc.analyze.find_outliers(
    df="single-cell-profiles.parquet",
    metadata_columns=[
      "Metadata_ImageNumber",
      "Image_Metadata_Plate_x"
    ],
    feature_thresholds={
      "Nuclei_AreaShape_Area": -1},
)
```

```
Number of outliers: 328 (19.14%)
Outliers Range:
Nuclei_AreaShape_Area Min: 734.0
Nuclei_AreaShape_Area Max: 1904.0

      Nuclei_AreaShape_Area  Metadata_ImageNumber Image_Metadata_Plate_x
23                    921.0                     2                Plate_2
28                    845.0                     2                Plate_2
29                   1024.0                     2                Plate_2
32                    787.0                     2                Plate_2
37                   1347.0                     2                Plate_2
...                     ...                   ...                    ...
```

Figure 2: ***The `find_outliers` function in coSMicQC uses single-cell feature thresholds to provide a report on how many outliers were detected (Python API or CLI). We use* z-scores *to help define thresholds used throughout coSMicQC.***

```
# CLI interface for coSMicQC find_outliers
$ cosmicqc find_outliers \
  --df single-cell-profiles.parquet \
  --metadata_columns \[Metadata_ImageNumber\] \
  --feature_thresholds '{"Nuclei_AreaShape_Area":
-1}'

Number of outliers: 328 (19.14%)
Outliers Range:
Nuclei_AreaShape_Area Min: 734.0
...
```

☆ 3) *Visualizing outlier distributions*

```
import cosmicqc
# label and show outliers within the profiles
scdf = cosmicqc.analyze.label_outliers(
    df="single-cell-profiles.parquet",
    include_threshold_scores=True,
).show_report()
```

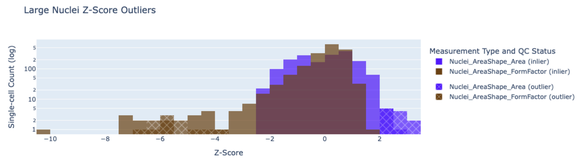


Figure 3: ***coSMicQC enables erroneous anomaly analysis through the `label_outliers` function, which appends z-score data for features, and the `CytoDataFrame.show_report` method to visualize where outliers are detected within the dataset.***

☆ 4) *Understanding outlier segmentations*

```
import cosmicqc

# passing image and mask dirs to display images
cosmicqc.CytoDataFrame(
    data="single-cell-profiles.parquet",
    data_context_dir="./image_directory/",
    data_mask_context_dir="./mask_directory/",
)
```

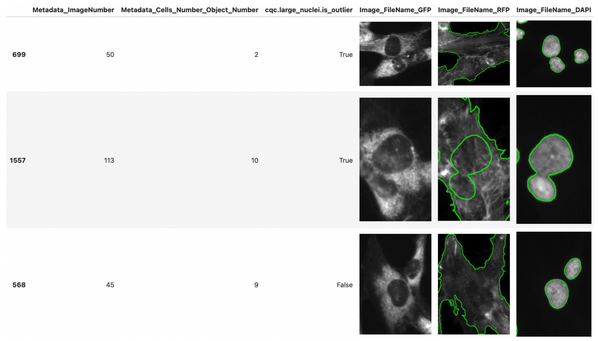


Figure 4: ***Interactive visualizations that help users identify outlier distributions through the* CytoDataFrame *— a novel data format that links single-cell measurements with their corresponding images and segmentation masks in real-time, enriching data analysis and interpretation.***

IV. Real-world applications

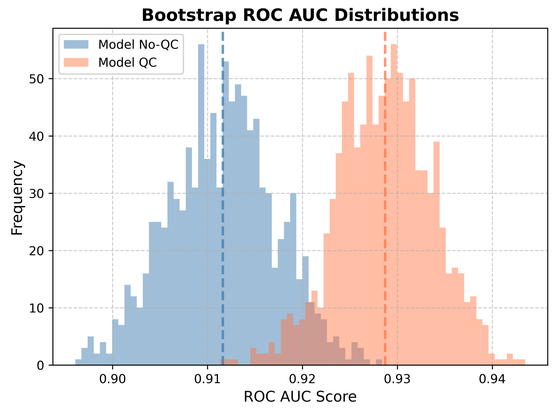


Figure 5: ***This figure displays the Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) scores for multiple random samples from a holdout dataset that has undergone quality control (QC). The ROC AUC scores are compared between models trained with QC (QC model) and those trained without QC (no-QC model). The QC model demonstrates superior performance, with consistently higher average ROC AUC scores compared to the no-QC model. Statistical analysis reveals a significant difference in performance, with a t-statistic of −72.1 and a p-value of 0.0, indicating that the QC model's enhancement is statistically robust. This highlights the effectiveness of applying QC to improve model accuracy and reliability.***

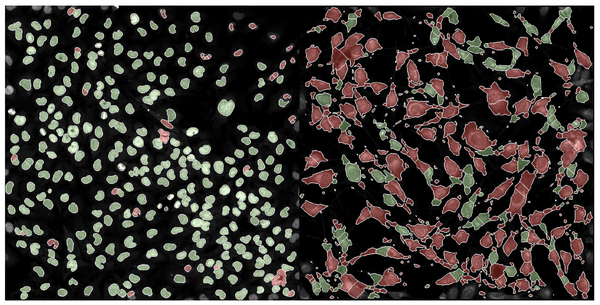


Figure 6: ***Single-cell segmentations (nuclei) were evaluated with* coSMicQC*, identifying which passed (green) or failed (red) quality control (QC) criteria. The left panel showcases field-of-view (FOV) images displaying nuclei from a more standard phenotype while the right panel shows nuclei from a sample with an unusual phenotype. These results illustrate how* coSMicQC *effectively distinguishes between high- and low-quality segmentations, aiding in the accurate identification of outliers and ensuring the reliability of downstream analysis for complex biological datasets.***

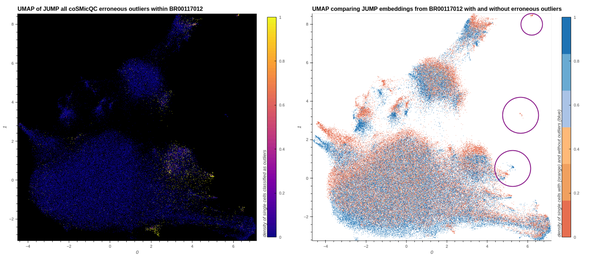


Figure 7: ***Applying* coSMicQC *to the JUMP dataset BR00117012 (cpg0000) reveals erroneous outliers, which are highlighted in yellow in the left panel. These outliers significantly impact the UMAP embeddings by altering the spatial distribution of data points. Specifically, the presence of outliers causes shifts in cluster locations or even their removal from the embeddings. In the right panel, orange points represent UMAP embeddings that include these outliers, while blue points denote embeddings generated after removing outliers. Some exemplary areas of significant change are circled in purple within the right panel.***

V. Acknowledgements

Special thanks goes to the following for their help in contributing to the **coSMicQC** inspiration, development, or related work.

- ***CU Anschutz CFReT***: Timothy A. McKinsey, Josh Travers
- ***iNFixion***: Michelle Mattson-Hoss, Herb Sarnoff
- ***Cold Spring Harbor Laboratory***: Katherine Alexander
- ***JUMP-Cell Painting Consortium***: Chandrasekaran et al., 2024 (cpg0000)
- ***St. Jude Children's Research Hospital***: Adam D. Durbin, Ha Won Lee, Taosheng Chen, and Noha Shendy