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

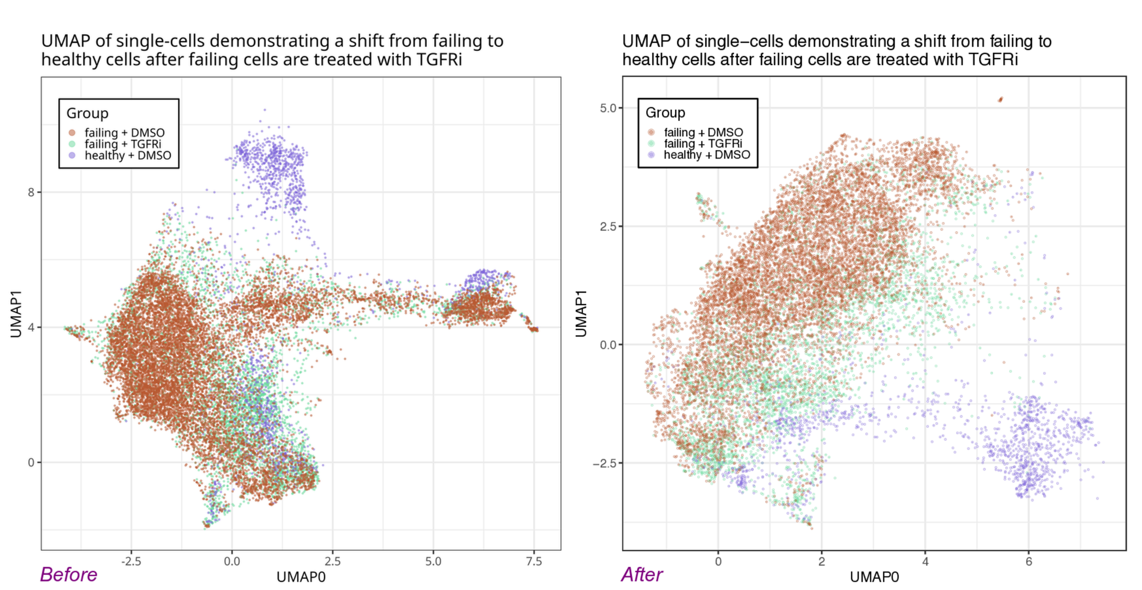
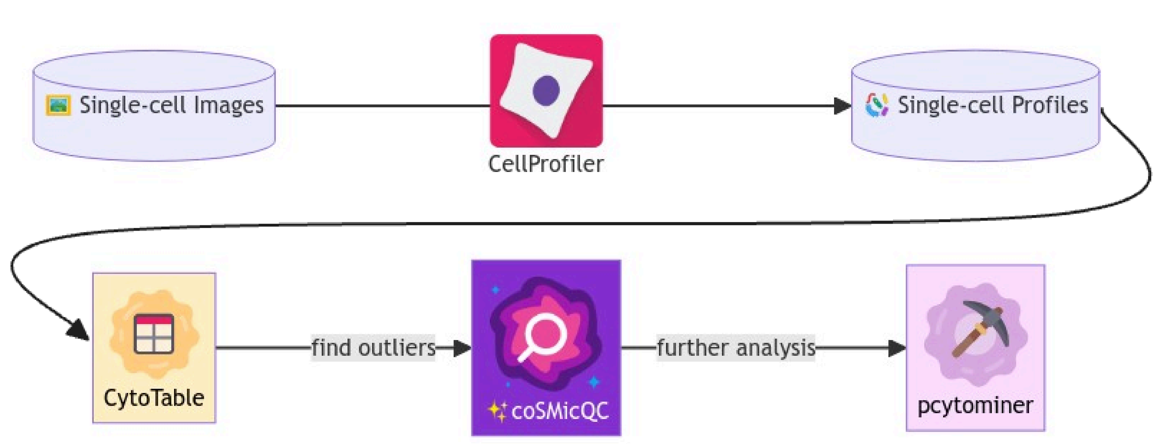


Figure 1: ***Extra clustering islands can be seen when looking at morphological profiles linked to poor segmentation, which when removed, better reveal patterns in the data.***

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. Researchers often resort to ***error-prone, bespoke filtering 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

# 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
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
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()
```

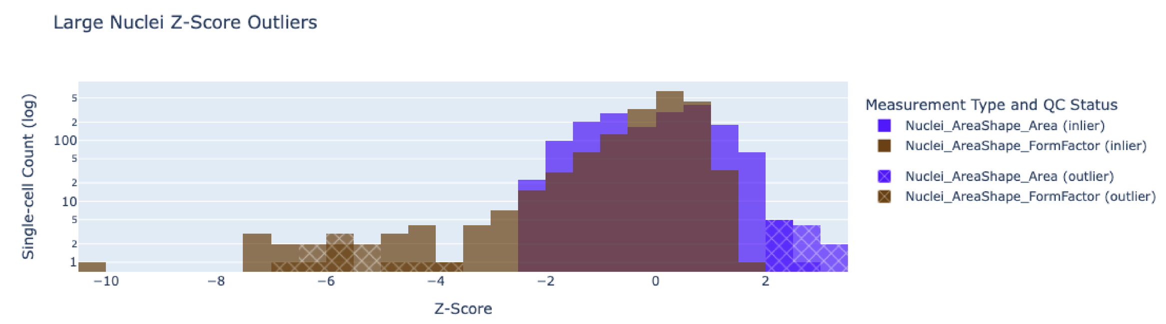


Figure 3: ***Deep erroneous anomaly analysis is enabled within coSMicQC through the `label_outliers` function, which appends z-score data for features, and the `CytoDataFrame.show_report` method to visualize where outliers are detected within the dataset.***

☆ 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/",
)
```

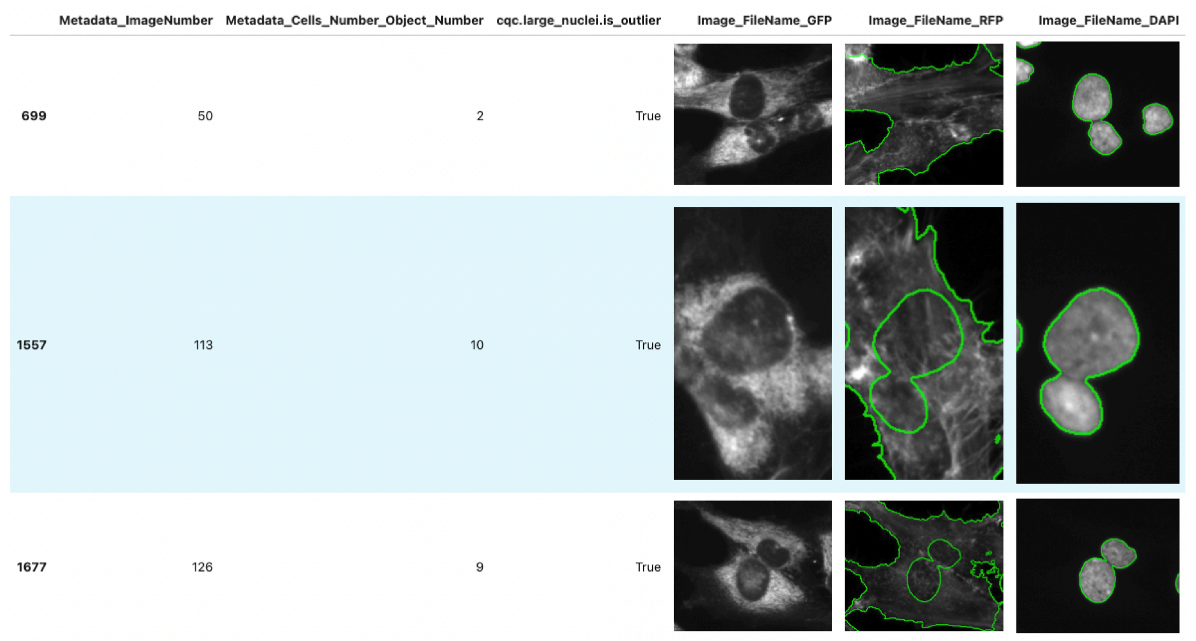


Figure 4: ***Interactive visualizations that help users identify outlier distributions through the CytoDataFrame — a novel data format that links single-cell measurements with their corresponding images and segmentation masks in real-time, enriching data analysis and interpretation.***

IV. Real-world applications

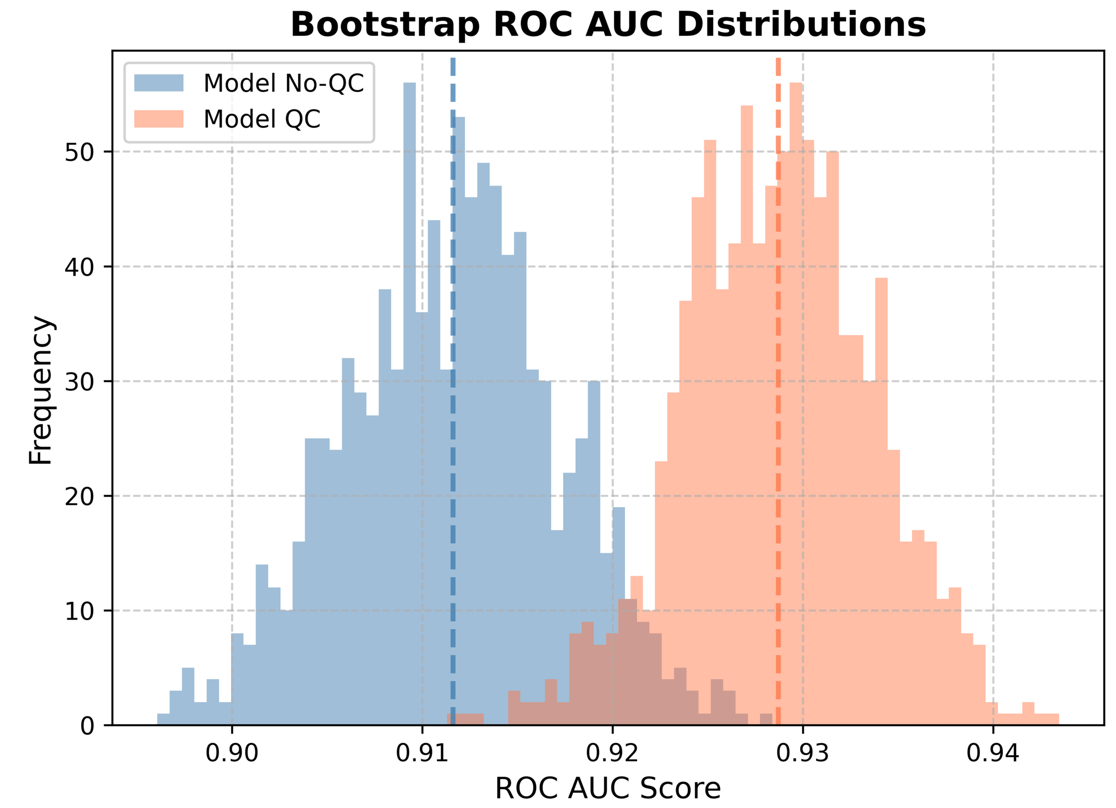


Figure 5: ***A representation of ROC AUC scores from multiple random samples taken from a holdout dataset (that has had QC applied) being applied to the QC trained model and no-QC trained model. The QC model outperforms the no-QC model, and the average performance of the QC model is significantly higher than the no-QC model (t-statistic = −72.1, p-value = 0.0).***

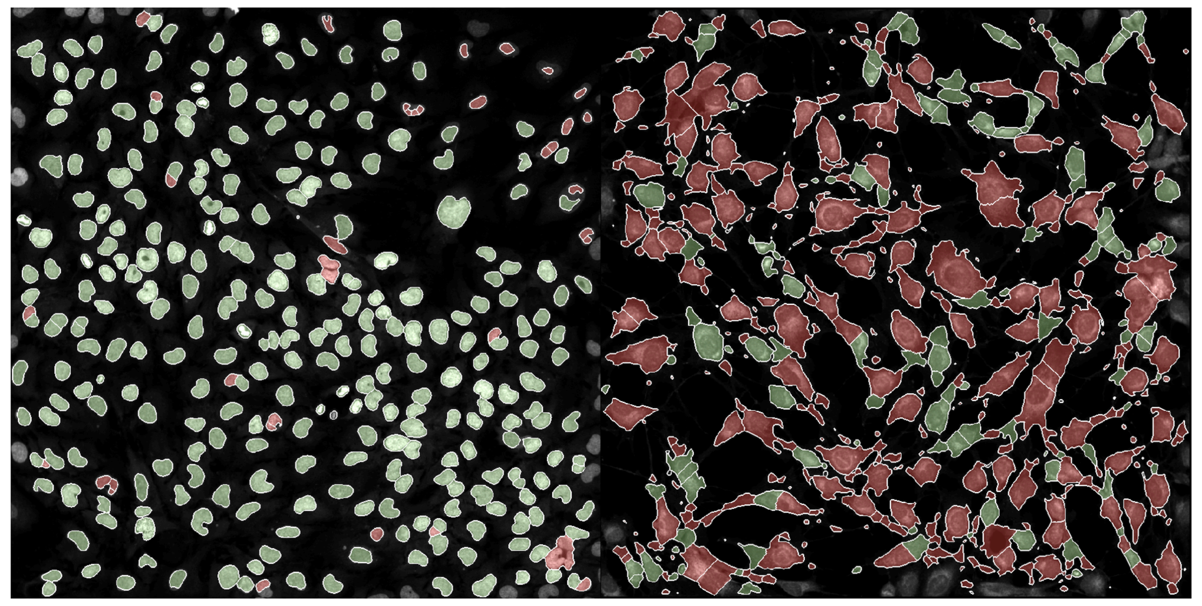


Figure 6: ***coSMicQC helped identify single-cell segmentations which passed (green) or failed (red) the QC conditions for two nuclei channel FOVs with an unusual phenotype (left) and a standard phenotype (right).***

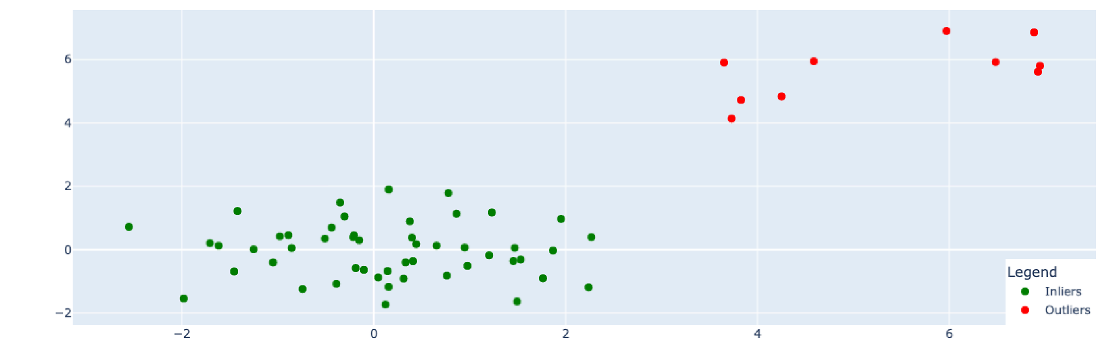


Figure 7: ***Praesent eu sem id nibh viverra finibus.Praesent eu sem id nibh viverra finibus.Praesent eu sem id nibh viverra finibus.Praesent eu sem id nibh viverra finibus.***

V. Acknowledgements

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

- ***CU Anschutz***: 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