METHODOLOGY **Open Access**

Toward generalizable phenotype prediction from single-cell morphology representations

Jenna Tomkinson[1†], Roshan Kern[1,2†], Cameron Mattson[1] and Gregory P. Way[1*]

Abstract

Background Functional cell processes (e.g., molecular signaling, response to stimuli, mitosis, etc.) impact cell phenotypes, which scientists can measure with cell morphology. However, linking these measurements with phenotypes remains challenging because it requires manually annotated labels. We propose that nuclear morphology can be a predictive marker for cell phenotypes that are generalizable across contexts.

Methods We reanalyzed a pre-labeled, publicly-available nucleus microscopy dataset from the MitoCheck consortium. We extracted single-cell morphology features using CellProfiler and DeepProfiler, which provide robust processing pipelines. We trained multinomial, multi-class elastic-net logistic regression models to classify nuclei into one of 15 phenotypes such as 'Anaphase,' 'Apoptosis', and 'Binuclear'. We rigorously assessed performance using F1 scores, precision-recall curves, and a leave-one-image-out (LOIO) cross-validation analysis. In LOIO, we retrained models using cells from every image except one and predicted phenotype in the held-out image, repeating this procedure for all images. We evaluated each morphology feature space, a concatenated feature space, and several feature space subsets (e.g., nuclei AreaShape features only). We applied models to the Joint Undertaking in Morphological Profiling (JUMP) data to assess performance using a different dataset.

Results In a held-out test set, we observed an overall F1 score of 0.84. Individual phenotype scores ranged from 0.64 (moderate performance) to 0.99 (high performance). Phenotypes such as 'Elongated', 'Metaphase', and 'Apoptosis' showed high performance. While CellProfiler and DeepProfiler features were generally equally effective, concatenation yielded the best results for 9/15 phenotypes. LOIO showed a performance decline, indicating our model could not reliably predict phenotypes in new images. Poor performance was unrelated to illumination correction or model selection. Applied to the JUMP data, models trained using nuclear AreaShape features only increased alignment with the annotated MitoCheck data (based on UMAP space). This approach implicated many chemical and genetic perturbations known to be associated with specific phenotypes.

Discussion Poor LOIO performance demonstrates challenges of single-cell phenotype prediction in new datasets. We propose several strategies that could pave the way for more generalizable methods in single-cell phenotype prediction, which is a step toward morphology representation ontologies that would aid in cross-dataset interpretability.

Keywords High content microscopy, Single-cell phenotype, Image-based profiling, Machine learning, CellProfiler analysis

[†] Jenna Tomkinson and Roshan Kern Co-first authors.

*Correspondence:
Gregory P. Way
gregory.way@cuanshutz.edu
Full list of author information is available at the end of the article

Introduction

Cell phenotypes are inherently dynamic, influenced by genetics, environmental factors, and intercellular interactions. These phenotypes change during important cell processes such as division, differentiation, disease, and death. Furthermore, scientists can induce phenotypic changes through chemical or genetic perturbation to uncover drug mechanisms or understand fundamental biological functions [1, 2]. These explorations often use a bioinformatics technique known as image-based profiling [3–6], which extracts cell morphology—unbiased cell state indicators of single-cell shapes, sizes, and intensity patterns—using feature extraction software, such as CellProfiler [7], DeepProfiler [8], and other bespoke methods [9, 10]. Despite these advances, accurately linking morphology to specific phenotypes poses a significant challenge, primarily due to the need for a priori annotation.

Researchers traditionally perform image-based profiling by aggregating every cell per well to create bulk profiles [11]. These bulk profiles overlook heterogeneity between single cells, but they eliminate outliers and make data more manageable. Bulk image-based profiling provides morphology information that describes important general readouts such as cell health, cell death, and chemical toxicity [12–14]. In contrast, single-cell morphology profiles provide an opportunity for single-cell phenotype prediction, which various groups have attempted. For example, Neuman et al. extracted 190 single-cell features and trained a support vector machine (SVM) to predict 16 single-cell phenotypes with 87% training set accuracy. [15] Additionally, Harder et al. trained an SVM with a Gaussian Radial Basis Function (RBF) kernel to predict four phenotype categories from nuclei images with 96% test set accuracy. [16] Other approaches incorporate time-lapse information, which models the likelihood of cell state transitions and improves performance [17–19]. Scientists have also applied deep learning to microscopy images directly to predict single-cell phenotypes (reviewed in Pratapa et al. [20]), most often using convolutional neural networks [21] or autoencoders [22]. However, these approaches do not rigorously test the generalizability of single-cell phenotype prediction in new datasets. Other approaches have successfully mapped bulk signatures across datasets, but these primarily focus on linking perturbation signatures rather than individual single-cell phenotypes [23–26]. In this work, we sought to overcome this challenge by developing an evaluation approach to test the generalizability of single-cell phenotype prediction across datasets. To maximize generalizability, we trained machine learning models using readily available and reproducible CellProfiler and DeepProfiler features to predict single-cell phenotypes from nucleus features alone.

We used the MitoCheck dataset, which includes nuclei imaging of HeLa cells manually labeled into one of 15 phenotypes [15]. We trained and extensively evaluated a multi-class elastic net logistic regression classifier through rigorous benchmarking. We found that our model could accurately predict phenotypes using traditional and deep learning feature extraction methods. The CellProfiler features slightly outperformed DeepProfiler features, but most top-performing models included both feature spaces. Despite achieving a high F1 score in held-out test sets for most phenotypes, our model performed remarkably poorly in a systematic leave-one-image-out (LOIO) analysis, which was not explained by illumination correction or model selection. We nevertheless modified and applied our approach to the publicly available JUMP Cell Painting dataset (CPJUMP1) [27]. We discovered that AreaShape features, and not those based on stain intensity, were resilient to dataset-specific biases. We predicted phenotypes in all CPJUMP1 single-cells and validated several perturbations with known phenotypic consequences. Overall, this work highlights the difficulties in generalizing single-cell phenotype predictions across datasets but suggests benchmarks and approaches to determine when effective generalization is achieved.

Results

Extracting morphology representations of phenotypically labeled nuclei

We analyzed a time-lapse fluorescence microscopy dataset called MitoCheck [15]. The data include GFP-tagged nuclei of HeLa cells perturbed with small interfering RNA (siRNAs) to silence approximately 21,000 protein-coding genes. The MitoCheck consortium's goal was to learn the mitotic function of genes by observing the mitotic consequences when they are knocked down. However, this work accomplished much more; it provided a publicly accessible microscopy dataset with high-quality annotations for 3,277 cells, each exhibiting one of 16 distinct phenotypes. It also contributed to growing cell phenotype resources, such as the Cellular Microscopy Phenotype Ontology [28], which provides API access to ontologies that link genes to phenotypes. After we acquired the MitoCheck data from Image Data Resource (IDR), a public repository hosting extensive microscopy datasets, we had annotations for 2,862 cells from 15 distinct phenotypes (dropping the 'Folded' phenotype) that are grouped into five distinct phenotype categories (Fig. 1A). We processed and analyzed this labeled data to train supervised machine learning models. Specifically, we applied image analysis, image-based profiling, and machine learning pipelines to process, extract, and analyze high-dimensional morphology features from

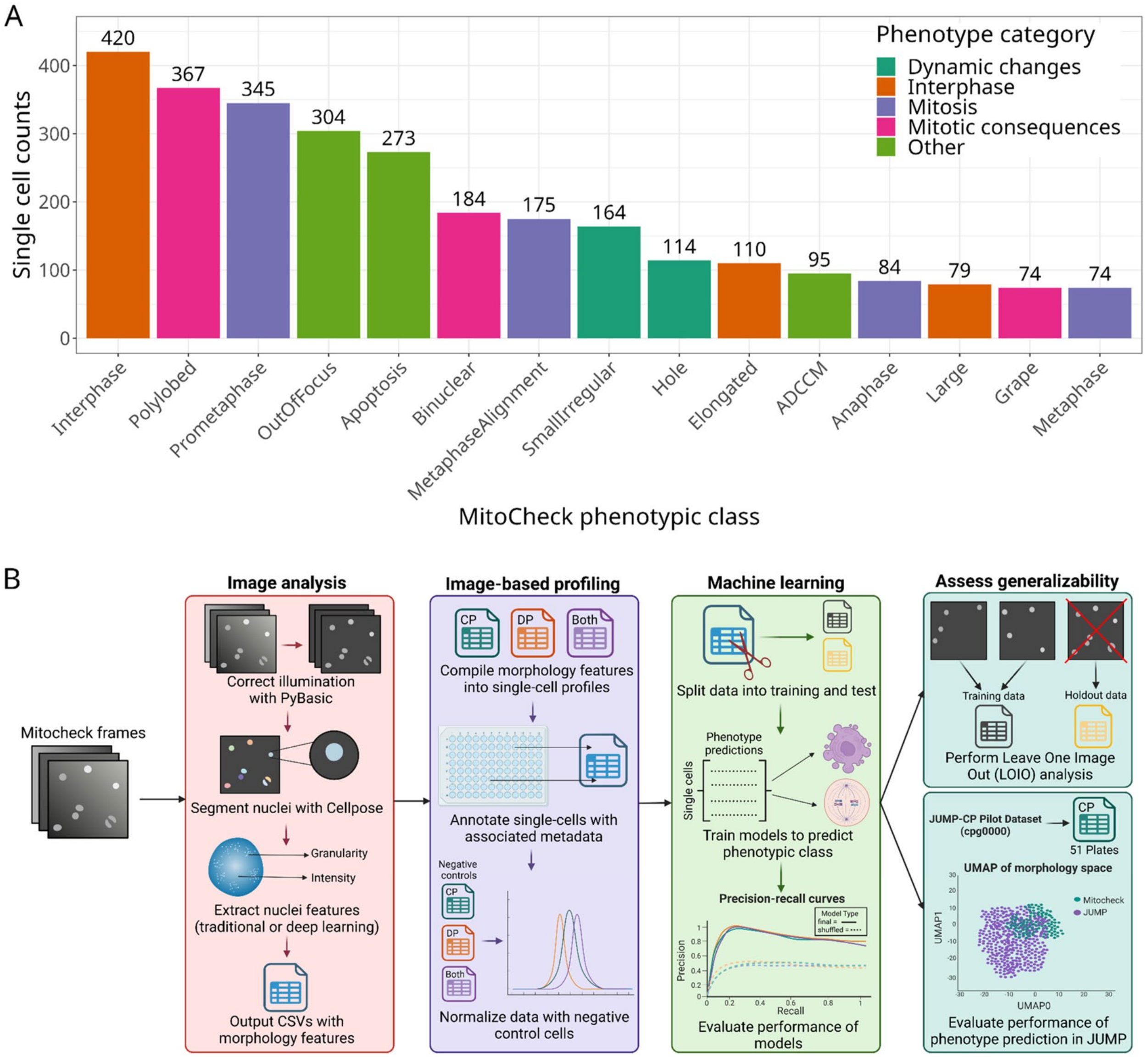

Fig. 1 *Dataset and analysis approach.* **A** Single-cell counts per labeled phenotype stratified by phenotype category. The labeled MitoCheck dataset included a total of 2,862 single nuclei. The original dataset contained labels for 16 classes, but we have removed "folded" because of low counts. **B** Our analysis pipeline incorporated image analysis, image-based profiling, and machine learning. We also assess model generalizability through a leave-one-image-out analysis and apply our models to the Joint Undertaking in Morphological Profiling Cell Painting (CPJUMP1) pilot dataset

MitoCheck nuclei to assess generalizable phenotype predictions (Fig. 1B).

We developed software called IDR_Stream to process MitoCheck data. IDR_Stream retrieves, and processes microscopy datasets directly from IDR, including MitoCheck [29]. IDR_Stream does not store raw data and other large intermediate files on disk, instead processing data in five steps: (1) temporarily downloading an image batch, (2) applying illumination correction with PyBaSiC [30], (3) segmenting nuclei with Cellpose [31], (4) extracting nuclei morphology features using both CellProfiler [7] and DeepProfiler [8], and (5) processing these morphology features using pycytominer [32] (Supplementary Fig. 1; see Methods for more details). We used IDR_stream to extract 157 nuclei morphology features using CellProfiler and 1,280 features using DeepProfiler from all 2,862 labeled nuclei. We also used IDR_stream to process 779,993 negative control nuclei from MitoCheck, which we used to normalize the 2,862 labeled nuclei. We selected these nuclei randomly to represent an expected distribution of all phenotypes; therefore, most are likely in interphase, but they are all without annotated phenotypes.

Evaluating heterogeneity of morphology feature spaces based on phenotypes

To broadly assess the relationships between single cells based on phenotypic class, we generated Uniform Manifold Approximation (UMAP) [33] embeddings from the nuclei morphology readouts from CellProfiler, DeepProfiler, and concatenated data (Fig. 2A). We found that for all feature datasets, 'OutofFocus' single cells (dark red) showed the most distinct islands. By eye, CellProfiler features demonstrated the most heterogeneity in UMAP space (more than DeepProfiler), particularly for select phenotypes (e.g., 'Elongated', 'Large', and 'Metaphase'). Other phenotypes were less distinct across all feature spaces (Supplementary Fig. 2). To quantify these observations, we calculated Silhouette scores [34] to report how much each feature space separated each phenotype. Indeed, CellProfiler had the highest Silhouette scores (0.53 total for all phenotypes), but DeepProfiler

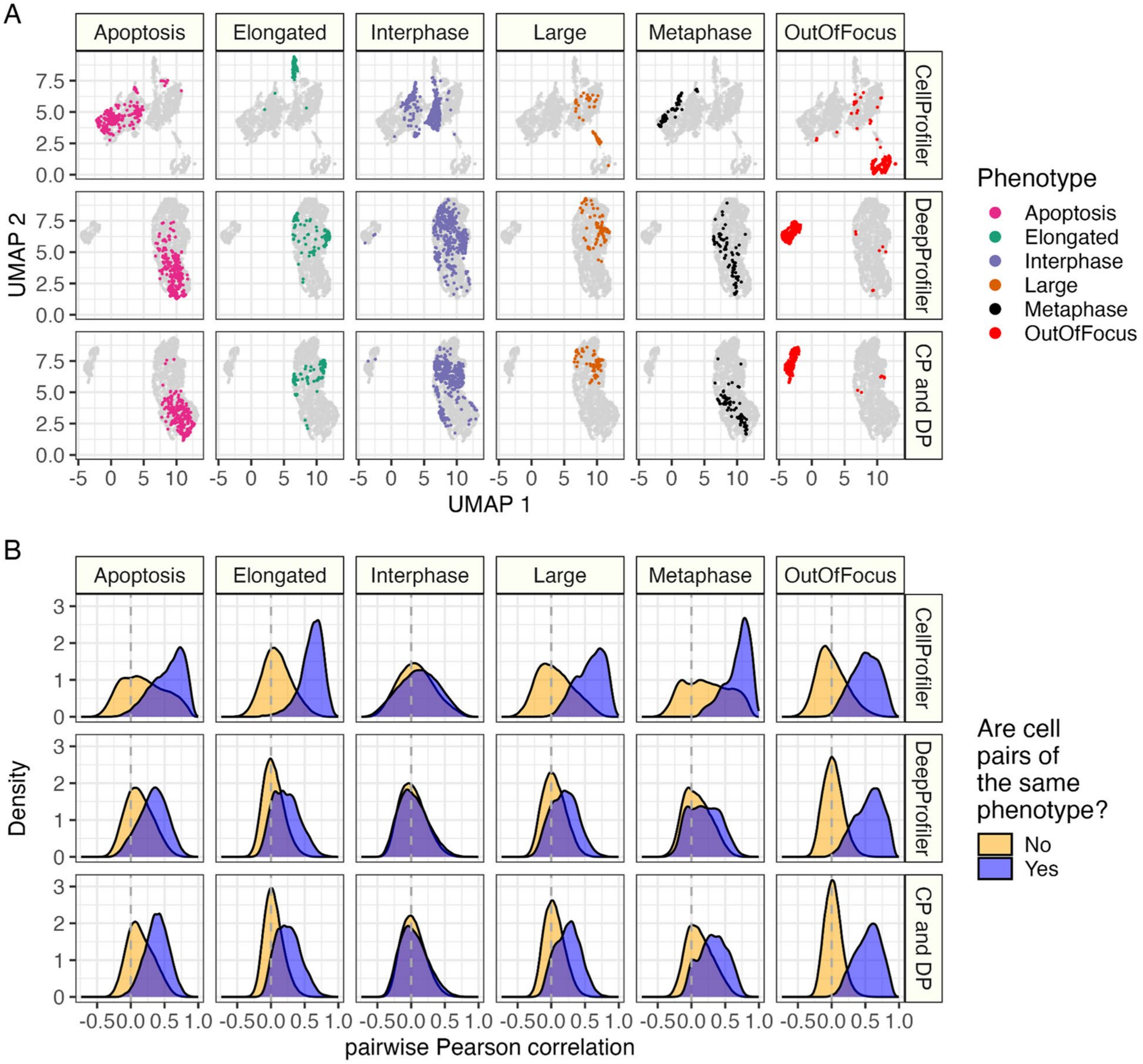

Fig. 2 *Some cell phenotypes are distinct, while others are more similar.* **A** Fitting three Uniform Manifold Approximations (UMAPs) per feature space (CellProfiler [CP], DeepProfiler [DP], and Combined [CP and DP]) shows distinct clustering of some but not all phenotypes. **B** Many phenotypic classes have highly correlated cell features, while others have low correlations compared to cells of different phenotypes

(0.18 total) had the highest scores for some phenotypes (Supplementary Fig. 3). We also generated embeddings using T-distributed Stochastic Neighbor Embedding (t-SNE) [35], which showed similar results as UMAP (Supplementary Fig. 4). Although the UMAP and t-SNE analyses indicated phenotype homogeneity, nuclei of the same phenotype had higher pairwise correlations than those of different phenotypes (Fig. 2B). However, interphase nuclei showed low pairwise correlations, likely due to normalization against negative controls containing mostly interphase nuclei (Fig. 2B). CellProfiler features showed the highest pairwise correlations of same-phenotype cells compared to cells of different phenotypes (Supplementary Fig. 5A). All other phenotypes showed variable but generally high pairwise correlations (Supplementary Fig. 5B). Based on these analyses, we expect that classifying most single-cell phenotypes is feasible but will likely only use a small subset of informative features.

Multi-class machine learning models classify single-cell phenotypes

We trained and rigorously evaluated multi-class machine learning models to predict the 15 single-cell phenotypes using single-cell morphology features extracted from MitoCheck data. We randomly split 85% of the data (evenly balanced by phenotype) into a training set and kept 15% as a test set for evaluation. We trained three independent models using each feature space individually (CellProfiler and DeepProfiler) and both feature spaces concatenated (CP and DP). We also separately trained “shuffled baseline” versions to serve as a random chance baseline in our evaluations.

Confusion matrices of the held-out test set data demonstrated strong performance across phenotypes (Fig. 3A). The performance based on precision-recall curves was also generally high, although the training set had nearly perfect performance, indicating some overfitting. The “shuffled baseline” models performed poorly, indicating that different class sizes or other technical artifacts did not bias our model training procedure (Fig. 3B). The combined CellProfiler and DeepProfiler dataset most accurately predicted phenotype for 9 out of 15 models and was top overall with an F1 score of 0.84 (Fig. 3C). CellProfiler features had top performance for 2/15 models (‘Interphase’, ‘Elongated’), while DeepProfiler features also had top performance for 4/15 different models (‘OutOfFocus’, ‘Large’, ‘Anaphase’, ‘ADCCM’). ‘ADCCM’ represents a phenotype class grouping artifacts, dynamic/folded, condensed, and other phenotypes [15].

We analyzed the machine learning coefficients from the multi-class models used to make phenotypic class predictions. The models generally used different features to predict each phenotype, indicating that most phenotypes can be explained by a unique set of nuclei measurements (Supplementary Fig. 6). We also trained and evaluated binary classification models to predict each phenotype individually, but these models demonstrated relatively poor performance in the test set compared to multi-class models (Supplementary Fig. 7). We expect this poor performance in our binary classification models to be driven by high phenotypic heterogeneity in the negative classes [36]. We therefore continue with a multi-class classification approach in subsequent analyses.

Leave-one-image-out analysis demonstrates poor generalizability

We performed a leave-one-image-out (LOIO) analysis to systematically test how our model generalizes to new images. Specifically, we retrained multi-class models using cells from every image except one, predicted single-cell phenotypes in the held-out image, and repeated this procedure for all 270 images (most images had many annotated single cells). While the test set performance was high (see Fig. 3), predictions in most individual images were poor. For each feature space, the prediction of the top-ranking phenotype by probability was often incorrect (Supplementary Fig. 8A). We observed, on average, correct phenotype predictions in only 22% to 26% of held-out images, with many phenotypes performing worse (Fig. 4A). In an attempt to minimize false positives, we set a high probability threshold ($p>=0.9$) for phenotype assignment, but we still observed many high-confidence incorrect predictions, albeit at lower proportions (Fig. 4B). Additionally, the incorrect predictions did not align with broader phenotypic categories (Fig. 4C). Poor LOIO performance was not a result of illumination correction, which we hypothesized could have introduced technical effects given our batched IDR_stream image processing, nor our decision to balance models by uneven class distributions (Supplementary Fig. 8B). Given that the LOIO images were collected in the same experiment, we doubted that models would generalize to new datasets collected in entirely different experimental settings. Nevertheless, we applied our models to the publicly available JUMP Cell Painting dataset (Joint Undertaking in Morphological Profiling; CPJUMP1) [27] to better understand model pitfalls and to work toward generalizable morphology annotation.

Single-cell phenotypic profiling in the CPJUMP1 dataset

The JUMP Cell Painting consortium released their pilot data (cpg0000) publicly. This dataset includes extracted CellProfiler features from perturbed A549 and U2OS cells with 303 chemical compounds, 335 Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) knockouts (targeting 175 unique genes), and

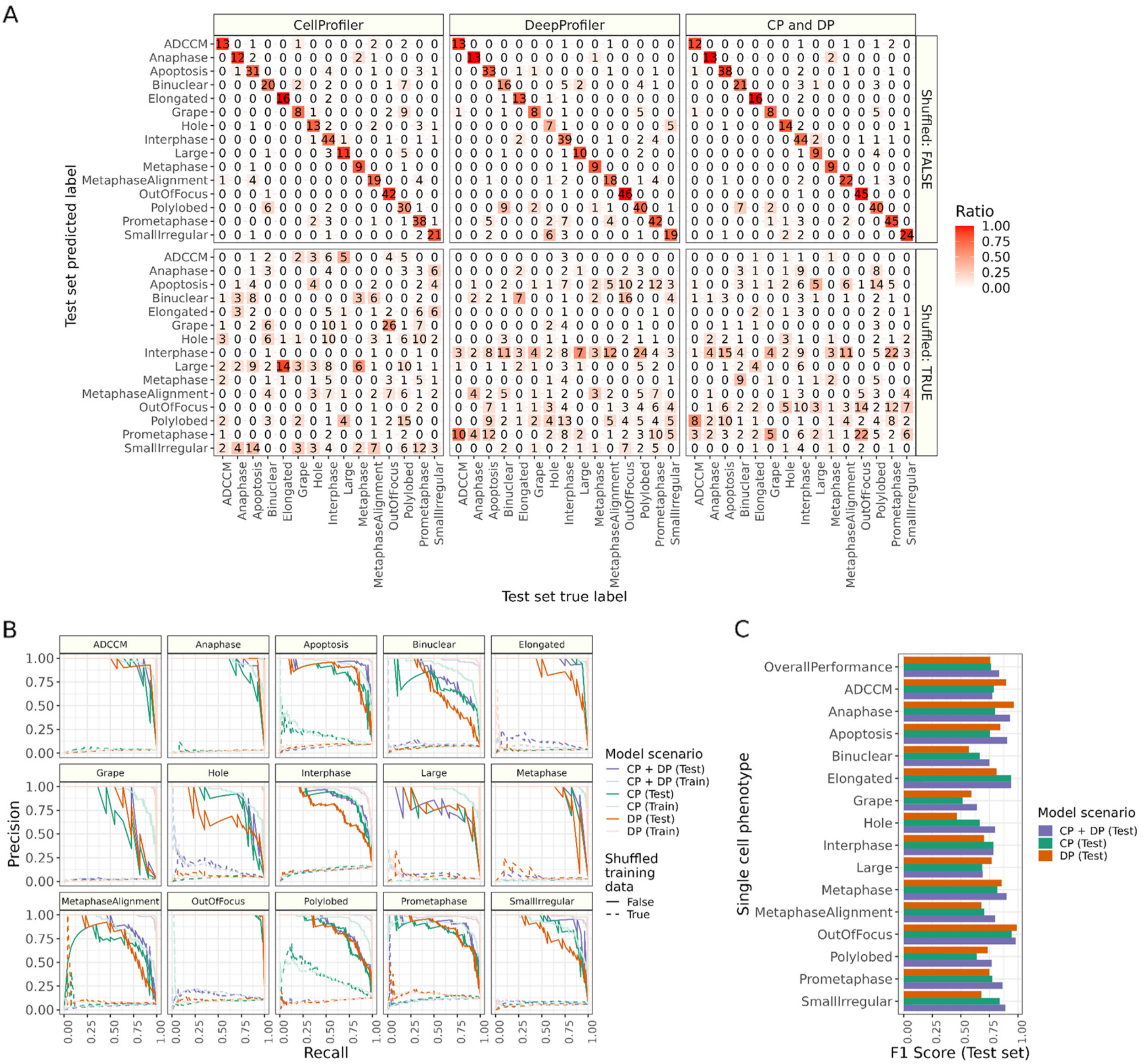

Fig. 3 *Evaluating multi-class predictions of single-cell phenotypes within the MitoCheck dataset.* **A** Confusion matrices comparing models trained on real data vs. shuffled data. The number in each box represents the total count, and the color represents the ratio of count over the ground truth label. All data show test set performance. **B** Precision recall curves for all 15 phenotypes. The shuffled baseline models (dashed line) performed poorly for all phenotypic classes. **C** F1 scores for test set predictions for 15 phenotypes and overall performance

175 overexpression open reading frame (ORF) reagents at two time points (short and longer incubation time) [27]. CPJUMP1 used the full Cell Painting panel, but we focused on analyzing the Hoechst nuclei stain to align with the MitoCheck GFP nuclei stain. We designed experiments to test if our phenotypic profiling model generalizes to data collected in an entirely different microscopy experiment in different cell lines with different stains collected over 15 years apart.

We began our investigation by aligning the MitoCheck features with CPJUMP1's precomputed CellProfiler nuclei features. Applying UMAP to this unified space demonstrated low sample overlap, suggesting large differences between the two feature spaces (Fig. 5A). We posited that technical parameters, including microscope acquisition and fluorescence staining, accounted for these observed differences. This would suggest that shape and area-based parameters are less affected by technical

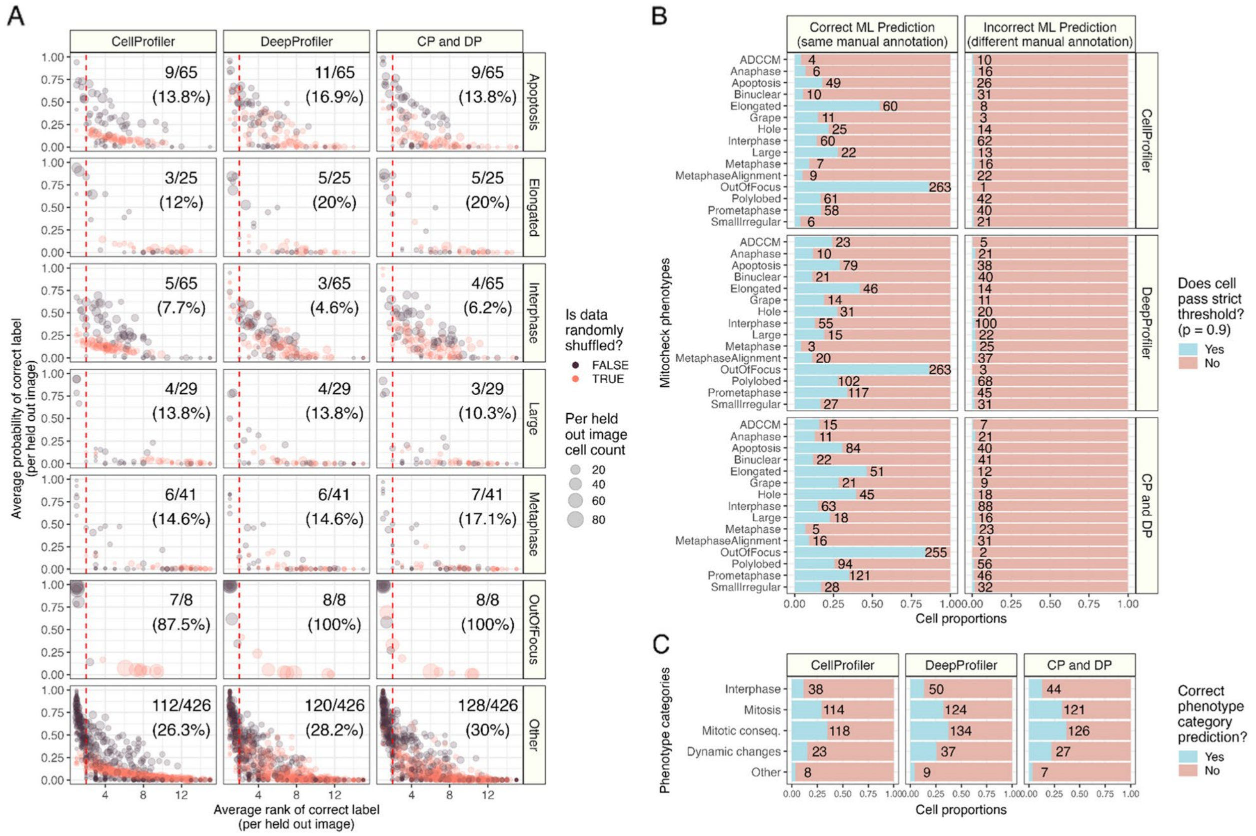

Fig. 4 *A Leave-One-Image-Out (LOIO) analysis demonstrated unexpectedly poor performance.* **A** Per-image LOIO results across feature spaces and select phenotypes. The dotted red line indicates rank two, indicating, on average, images with accurate phenotype labeling (most images have more than one annotated single cell). The text represents the number of images in the LOIO left out set with average predictions below rank two (high performance) for a given phenotype. **B** Setting a high probability threshold (p > 0.9) for calling single-cell phenotypes does not improve prediction reliability. The left set of bars indicates single cells that our model predicted correctly, while the right set of bars indicates incorrect single-cell predictions. Even for incorrect predictions, many individual single cells still pass a strict probability threshold (e.g., we incorrectly predicted ten cells as ADCCM phenotype using CellProfiler feature space with $p > 0.9$). **C** Performance does not improve even if we collapse predictions to phenotype category (MitoCheck assigned individual phenotypes to five distinct categories)

variations and better facilitate data integration. We systematically tested all CellProfiler features to identify which features were most different between the two datasets and confirmed that AreaShape features are the least different (Supplementary Fig. 9A). Within AreaShape features, we noted that Zernike features had the lowest difference (Supplementary Fig. 9B) and shared similar variance between datasets (Supplementary Fig. 9C). After dropping all other features (intensity-based features) and applying UMAP again, we observed a higher dataset overlap (Fig. 5B).

We retrained our multi-class logistic regression classifier using two additional feature subsets: AreaShape features and Zernike features only. As expected, we observed a drop in performance in predicting single-cell MitoCheck phenotypes, particularly for models trained using Zernike features only (Supplementary Fig. 10). However, given the misalignment of other features, we applied the higher-performing AreaShape model to all 20,959,860 single cells in the CPJUMP1 pilot. This procedure annotated all CPJUMP1 single-cells to phenotype probabilities.

Per phenotype and plate, we compared phenotype prediction probability distributions of negative controls to each treatment. We applied a two-sample Kolmogorov–Smirnov test (KS test) to these distributions to determine the enrichment of specific phenotypes in specific treatments compared to negative controls. We repeated this procedure for negative control models trained with randomly shuffled features. This resulted in 485,370 comparisons. We report the top 100 most enriched treatments per phenotype in Supplementary Table 1 and the full list in our GitHub repository. Generally, we observed higher phenotype enrichments for compound treatments than CRISPR or ORF, and shorter-duration incubation displayed a higher divergence across cell types

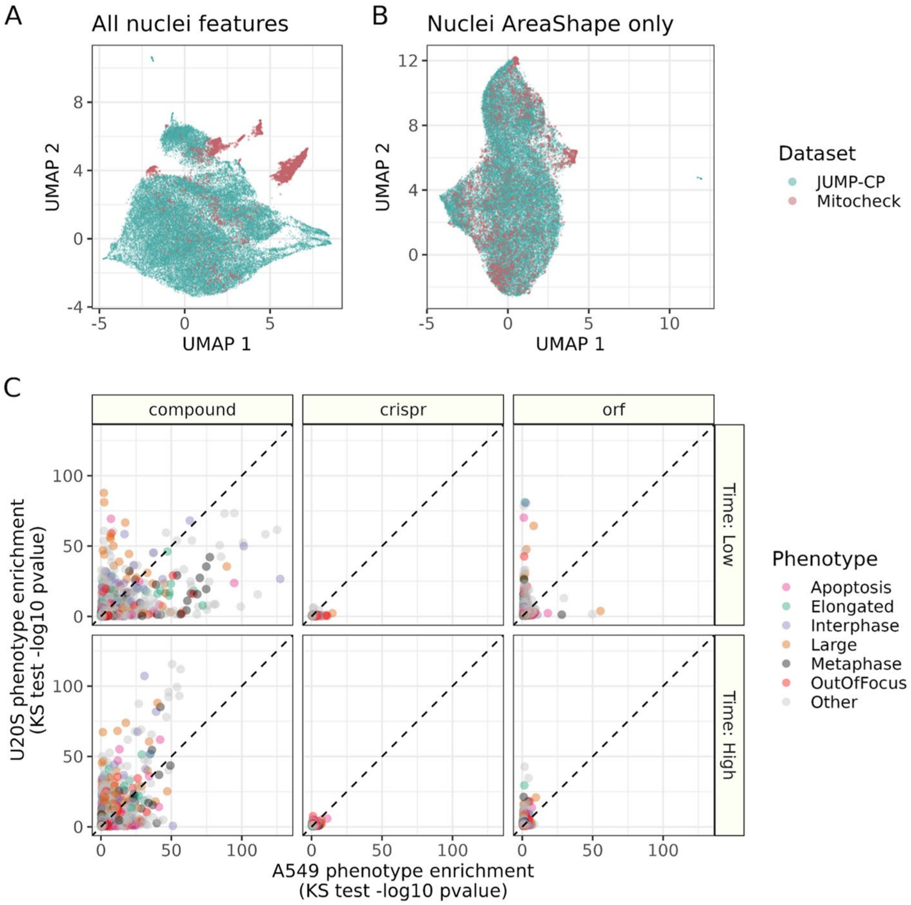

Fig. 5 *Investigating feature alignment and phenotype enrichment in the CPJUMP1 dataset.* UMAP projections of **A** combined MitoCheck and **(B)** CPJUMP1 feature spaces. The left panel represents all nuclei features, while the right panel includes features only belonging to AreaShape CellProfiler categories. **C** Comparing KS-tests between U20S and A549 for three treatment categories and two incubation periods. Only select phenotypes highlighted here; see Supplementary Fig. 9 for a full comparison of all phenotypes

compared to longer-duration incubation (Fig. 5C). Shuffled model enrichment was much lower than the ground truth model (Supplementary Fig. 11A). Large phenotypes generally had the highest enrichment, but no other phenotype displayed substantially elevated scores compared to other phenotypes, and most treatments did not have enriched phenotypes (Supplementary Fig. 11B-C). When analyzing individual treatments per phenotype, we identified fludarabine-phosphate, amino purvalanol-a, and *RPL23A*knockdown as significantly enriched for 'Apoptosis' phenotypes. Results for all three treatments have been reported previously [37–39]. Furthermore, oxibendazole, colchicine, and CYT-997 all showed enriched 'Elongated' phenotypes, which have been previously observed [13, 40–42]. This analysis provides a comprehensive statistical estimation of phenotype enrichment for the compound, ORF, and CRISPR treatment in the CPJUMP1 dataset, which can be mined for future hypothesis testing and perturbation annotation.

Discussion

Using publicly available data from the MitoCheck consortium, we show that high-content morphology features derived using classical computer vision techniques (CellProfiler) and deep learning approaches (DeepProfiler) effectively capture single-cell phenotype information from nuclei imaging. In held-out test sets, simple machine learning models reliably predicted 15 distinct single-cell phenotypes, ranging from apoptosis to specific mitotic phases and alternative nuclear forms like polylobed and grape. Initially, we aimed to apply these models to analyze unseen data to add single-cell phenotyping as an interpretation layer to any high-content microscopy experiment that marks nuclei (such as functional genomic and drug discovery screens). However, we encountered significant challenges in our leave-one-image-out (LOIO) analysis; a process that systematically retrained phenotype predictors on all single-cell data while excluding data from one specific image at a time. This analysis revealed that our models struggled to accurately predict single-cell phenotypes in individual images not used in training, which was not explained by analysis parameters such as illumination correction or machine learning model balancing. While we observed much better single-cell phenotype predictions compared to random guessing, poor LOIO results suggested that our approach will especially struggle with single-cell prediction in new datasets beyond MitoCheck. Nevertheless, we investigated how to apply our phenotype model to other publicly available microscopy data to better understand model behavior and pitfalls.

Analyzing the CPJUMP1 pilot data [27], we identified poor dataset alignment with MitoCheck as a pivotal issue. Aligning these datasets is complicated by differences in technical parameters (e.g., staining methods, microscopy techniques) and biological parameters (e.g., cell lines, treatments). [43] Despite these hurdles, we noted that certain features, like AreaShape, exhibited greater consistency across datasets, whereas other morphology features based on intensities displayed significant variability. This variability highlights the challenges of accurately annotating phenotypic signatures in unseen data and underscores the necessity of careful dataset alignment, including batch effect correction [44–46], to enhance the generalizability of image-based profiling. Applying a re-trained model using AreaShape features only to the CPJUMP1 data identified many treatments enriched for specific phenotypes that have already been observed. Coupled with low enrichment scores in our negative controls, our results show initial promise in this approach to assign phenotype to individual perturbations in external datasets. We do not perform a comprehensive investigation of all CPJUMP1 treatments in this paper, but we do provide a full list of all phenotype enrichment scores. We will apply this approach to the full JUMP dataset once its quality is controlled and released. In 2010, MitoCheck performed a similar approach, training an SVM to predict the phenotypic consequences of genome-wide siRNA knockdowns on cell function [15]. While the models they trained, on average, performed well in the labeled dataset, our analysis suggests that many individual single cells—and consequently, numerous genes—that were outside their training set, may have inaccurate annotations. However, while single-cell predictions struggle with generalizability, averaging single cells over many images to form bulk profiles likely improved MitoCheck phenotype-to-gene annotation, which is also what we may have observed in our CPJUMP1 analysis. In other words, our in-depth analysis of single-cell generalizability may point to a broader issue in our field, and while averaging single cells at least partially mitigates this issue, future research is needed to improve single-cell phenotype prediction across datasets.

Our overarching goal was to identify generalizable single-cell morphology signatures of phenotypes. Given the challenges and high time and labor costs associated with manually labeling phenotypes, integrating a pre-labeled dataset with unlabeled datasets could enable cheap and fast predictions for any unlabeled data. However, more research is needed to identify the best approach. A rigorous evaluation of individual CellProfiler morphology features could enhance dataset alignment and future phenotype annotation. Specifically, these features could be assessed for their stability across variations in illumination correction, segmentation, image rotation, and the presence of imaging artifacts like blur and saturation. Additionally, investigating technical parameters (such as different microscopes, stains, cell lines, and software versions) would improve our understanding of feature sensitivity. While DeepProfiler (and other deep learning feature extractors) might also identify generalizable single-cell morphology signatures, frequent updates to these models introduce the need for continuous data reprocessing. Batch effect correction is already a pivotal strategy for microscopy data alignment. [44, 45, 47, 48] A feasible approach may involve first aligning a labeled dataset with unlabeled data, retraining phenotype predictors in this harmonized space, and then deploying models for phenotype prediction in the unlabeled data. Alternatively, foundation models potentially offer consistent feature representation extraction across datasets, which would circumvent the alignment step [10]. For example, initiatives like "Mitospace", which focuses on extracting a common feature space of mitochondria [49], the masked autoencoder Phenom-Beta, which is a vision transformer foundational model for embedding microscopy images [10], “BioMorph”, which links morphology to organelle processes [50], and the Allen Cell Explorer, which uncovers cell phenotypes organelle-by-organelle [51, 52], illustrate promising future directions for annotating universal cell representations with generalizable single-cell phenotypes. Nevertheless, these foundation models still require phenotype interpretation to analyze cells and perturbations on a uniform, biologically interpretable basis. Lastly, collecting new, labeled datasets that span multiple cell lines and image acquisition parameters (e.g., different microscopes) will provide more information to improve and refine machine learning phenotype predictors. Taken together, all aforementioned efforts are likely required to solve this challenge, which, once addressed, will enable robust phenotype prediction and improve mechanistic annotations in future microscopy datasets.

Methods

MitoCheck data, labels, and quality control

Neumann et al. originally collected and used MitoCheck data for phenotypic profiling of cells perturbed with siRNAs targeting human protein-coding genes. [15] This dataset contains the raw timelapse data of live-cell HeLa nuclei imaged via the H2B protein tagged with GFP. While the manually annotated cell dataset used in Neumann et al. was compiled in 2007, the MitoCheck consortium continued to create manually annotated datasets until 2015. We used the most recent dataset to train and evaluate the models. The most recent MitoCheck-generated labeled dataset includes the phenotypic class label and location data for 3,277 cells. [17] The phenotypic class label is one of 16 classes (large, metaphase, apoptosis, etc.). While the original dataset contained 16 phenotypes, we dropped "folded" due to low sample counts. Location data for a cell includes its respective plate, well, frame, and centroid coordinates.

MitoCheck consortium pre-preprocessed the mitosis movies using a two-step quality control (QC) procedure based on automatic and manual data inspection [15]. MitoCheck applied this procedure before uploading to Image Data Resource (IDR) [29]. Therefore, we did not use any data that failed the original QC. We performed an additional round of QC by inspecting illumination artifacts. We discarded frames from well A1 from each plate, as we observed consistently irregular illumination, with each of these wells having significantly darker illumination in the center of the frame (Supplementary Fig. 12). This differed significantly from the vignetting observed in other wells, leading to errors in our illumination correction during preprocessing. After removing cells that failed QC and "folded" cells, 2,862 cells remained as our final analytical set.

Downloading MitoCheck data with IDR_Stream

We developed IDR_Stream to rapidly acquire and process public microscopy datasets with low computational overhead. Image analysis and image-based profiling pipelines typically produce gigabytes to terabytes of intermediate files related to each step of the pipeline [5]. For our pipeline, these files included raw images, preprocessed images, segmentation masks, and image-based morphology profiles in various intermediate data processing formats (i.e., annotated, normalized, feature selected) [32]. If compiled single-cell features are the only necessary data for downstream analyses, intermediate data can unnecessarily clog large amounts of machine storage space. IDR_stream therefore downloads the public raw images, performs illumination correction, segmentation, feature extraction, and image-based profiling processing (Supplementary Fig. 1). Importantly, this tool processes images in batches, deleting unnecessary files after completing each batch. We used IDR_Stream to access the MitoCheck raw images, which is the only form of data provided.

We used IDR_Stream to access the MitoCheck raw images. Given a metadata input file that includes the location data (plate and well) of MitoCheck movies, IDR_Stream uses Aspera high-speed transfer client to download the MitoCheck images from IDR (accession: idr0013-screenA). We download the files in CellH5 format, an HDF5 data format for cell-based assays [53]. Each CellH5 file contains 93 frames of live cell imaging data.

Applying illumination correction with IDR_Stream

IDR_Stream uses Bio-Formats to read the CellH5 format. Bio-Formats bypasses the need for format conversion by reading image data directly from proprietary formats [54]. IDR_Stream uses PyImageJ to access Bio-Formats with Python. Rueden et al. created PyImageJ as a bridge between Python and ImageJ. [55] IDR Stream uses the BaSiC method for illumination correction of each well [30]. We use the Python implementation of the BaSiC method, named BaSiCPy. We used the default BaSiCPy parameters for illumination correction. The BaSiC method works well for preprocessing time-lapse data, accounting for time-lapse-specific illumination artifacts such as photobleaching. BaSiCPy requires at least three images to perform illumination correction. We provide BaSiCPy with two frames before/after the desired frame (depending on its position in time). After illumination correction, IDR_stream keeps only the frame of interest for further processing.

Segmenting nuclei with IDR_Stream

IDR_Stream uses the Python implementation of the CellPose segmentation algorithm to segment the nuclei in each mitosis movie [31]. The CellPose segmentation models were trained on a diverse set of cell images, and the Python implementation was particularly useful for building reproducible pipelines. We manually experimented with CellPose on ten images to determine the optimal CellPose parameters for segmenting nuclei. Manual experimentation involved examining nuclei segmentation across each image to ensure they looked as expected. Ultimately, we used the CellPose cytoplasm model for segmentation, which we found segmented nuclei in MitoCheck significantly better by eye than nucleus models. We used a diameter size of 0, which requires the CellPose model to estimate nuclei diameters for each image. We also increased the flow threshold parameter from its default value of 0.4 to 0.8, which increased the maximum error allowed for the flow of each cell mask. We found that CellPose could not segment some nuclei without increasing the flow threshold parameter. We also remove nuclei masks on the edge of an image to avoid capturing partial nuclei information.

Extracting and processing morphology features with IDR_Stream

IDR_Stream uses CellProfiler and DeepProfiler to extract features. We use CellProfiler version 4.2.4 to extract all features within the following categories: granularity, object intensity, object neighbors, object intensity distribution, object size shape, and texture [7]. The CellProfiler output is a CSV file with single cell metadata and features. DeepProfiler extracts morphological features using a pre-trained convolutional neural network and weakly supervised learning [8]. This model extracts features from five Cell Painting channels (DNA, ER, RNA, AGP, Mito). We repurposed the model to extract features from the MitoCheck mitosis movies as DNA channel features only. We also changed parameters in the DeepProfiler software. Specifically, we changed the label of interest from "Allele" to "Gene" because of the siRNA perturbations in MitoCheck. We also changed the box size parameter from 96 to 128 to increase the context around each cell. We used the DeepProfiler GitHub hash version: 2fb3ed3027cded6676b7e409687322ef67491ec7. IDR Stream can optionally concatenate single-cell features extracted by both CellProfiler and DeepProfiler from each batch into a single data frame, which includes metadata. IDR_Stream uses pycytominer to compile and annotate the single-cell embeddings extracted using either CellProfiler and/or DeepProfiler [32]. Importantly, we also applied IDR_stream to process MitoCheck negative control nuclei, which we used for normalization. These nuclei represent an expected distribution of all phenotypes present in the data, but about 95% of nuclei are likely in interphase [56]. This procedure of using all negative controls is robust in the presence of dramatic phenotypes [5]. Specifically, we learned z-score normalization parameters from all negative control cells (per feature) and applied this transformation to all MitoCheck cells (including those with annotated phenotypes). Sklearn standard scaler standardizes features by removing the mean and scaling to the unit variance of negative control cells [57].

Formatting the MitoCheck labels: accounting for IDR_stream processing differences

We expect some minor differences between MitoCheck processing and IDR_stream processing. Specifically, IDR_stream identifies centroid coordinates for every segmented nuclei, and these coordinates are likely different from MitoCheck centroids because of different segmentation parameters. MitoCheck does not provide segmentation masks for us to confirm or quantify these differences. We nevertheless must assign the correct MitoCheck phenotypic class labels to the appropriate IDR_Stream processed cells. We therefore used IDR_stream-based masks (segmentation outlines) to assign MitoCheck phenotypes. If a MitoCheck centroid was within the IDR_stream mask, then we assigned that IDR_Stream nuclei the given MitoCheck phenotype. In practice, we do not expect this impacted single-nuclei phenotype assignment.

Data splitting and machine learning training procedures for phenotype prediction

We randomly split 15% of the MitoCheck labeled dataset into a separate held-out test set balanced by phenotypic class. We used the remaining 85% to train all phenotypic profiling models. We used 2,432 samples as the training set and 430 as the test set. We trained logistic regression models with elastic net penalty using sklearn version 1.1.1 [57]. This model is computationally efficient, easily interpretable, and induces sparsity in selecting model features. To understand how different feature sets affected the models' performances, we used three different feature types to train and test each model: CellProfiler features (CP), DeepProfiler features (DP), and a combination of these features (CP and DP). To produce a suitable baseline for generalizable performance, we repeated the steps to train final models with randomly shuffled data. In the shuffling procedure, we randomly shuffled the features independently per column before training. There were two *model types* for each logistic regression model: final and shuffled baseline. We evaluated the shuffled model on non-shuffled test set data.

We trained two forms of models: 1) multi-class, single-label models and 2) binary classification models. The multi-class models predict a probability for each of the 15 phenotypic classes given a vector of features per single cell. We used a multinomial multi-class training procedure. The binary classification models predicted a probability for "positive" or "negative" for its respective phenotypic class. After an initial evaluation, we also trained multi-class models using only AreaShape and Zernike features with the class_weight parameter in sklearn specified as "balanced". In total, we trained 20 multi-class logistic regression models (2 class_weight types * 5 feature types * 2 shuffle types). Since each phenotypic class had a specific binary classification model, there were 90 binary classification models (3 feature types * 2 shuffle types * 15 phenotypic classes). In total, we trained and evaluated 110 phenotypic profiling models.

We performed a grid search and ten-fold cross-validation on each model using the training subset to identify optimal regularization and elastic net mixing parameters. We tested for cross-validation performance using seven different regularization parameters ([1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03]) and ten different elastic net mixing parameters ([0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]). The regularization parameter controls the penalty term for all features, and the elastic net mixing parameter controls the trade-off between L1 and L2 regression (0 = L2 and 1 = L1). Therefore, the closer the elastic net mixing parameter is to 1, the sparser the model. We set the model scoring to "F1 weighted", meaning that the model tries to maximize the average weighted by supporting the F1 score across the training data.

For the binary classification models, we downsampled negative labels to get an even split of positive and negative training labels from the training data. For example, if there were only 50 positive labels for a particular class, we would randomly sample the negative labels to create a training set with 50 negative labels. Undersampling helps reduce the bias inherent in datasets with the most negative labels. We trained multi-class models with the full training dataset.

Assessing feature space representations

We quantified the ability of each feature space to separate single nuclei into phenotype clusters. We applied Silhouette score [34], comparing each individual phenotype to all other nuclei for each feature space. Briefly, Silhouette scores quantified phenotype representation tightness and separation compared to all other phenotypes [34]. High Silhouette scores tell us which phenotypes have tightly clustered nuclei, while low Silhouette scores tell us which phenotypes look like other phenotypes. Prior to calculating Silhouette scores, we transformed each feature space using Principal Components Analysis (PCA) with 50 principal components. This procedure ensures that we capture a high proportion of variance for each feature representation while standardizing the dimensionality of Silhouette score calculations.

Evaluating phenotype prediction performance

After training, we evaluated the 110 phenotypic profiling models with F1 score, precision-recall curves, and confusion matrices. The F1 score metric included an F1 score for each phenotypic class present in a model (positive/negative) and a weighted F1 score. The F1 scores measure the models' balanced precision and recall performance for each class, weighted by the number of true instances for each class. The precision-recall curves show the tradeoff between precision and recall for different classification thresholds. Confusion matrices illustrate the models' true and false positive and false negative predictions.

We also performed leave-one-image-out (LOIO) training and prediction for each phenotypic profiling model. For each target image in the MitoCheck labeled dataset, we use the cells not from the target image to train the multiclass model as described above. We then use this trained model to predict phenotype probabilities for each cell from the left-out image. LOIO evaluation shows how well the model will perform on cells from an image the model has never seen before.

Interpreting phenotype models

Generally, the coefficients of the models correspond to how the model makes use of specific features in predicting a phenotypic class, where a positive value means a feature is generally more likely to contribute to the corresponding class, and a zero value means the feature does not contribute to the class's predicted probability. We applied hierarchical clustering and visualization of logistic regression coefficients using ComplexHeatmap [58].

Accessing CPJUMP1 pilot data

We accessed the Broad Institute's publicly available Cell Painting data from the JUMP-Cell Painting Consortium [27]. We analyzed the CPJUMP1 pilot dataset, which consists of 51 plates with approximately 21 million cells. The public release includes CellProfiler cell morphology features of three perturbation categories (ORF, CRISPR, and compound) across two cell lines (A549 and U2OS). We accessed these CellProfiler features (SQLite files) from the Cell Painting Gallery [59] (accession number cpg0000), which is a public Amazon Web Services (AWS) S3 bucket. We accessed the corresponding platemap and metadata manifests from the JUMP GitHub repository. We provide a guide to access these data at https://github.com/WayScience/JUMP-single-cell/tree/main.

Processing CPJUMP1 pilot data

We processed the CPJUMP1 pilot dataset from the public SQlite plate files of CellProfiler features using pycytominer [32]. As noted in the CPJUMP1 manuscript, the CellProfiler version was either 4.0.7, 4.1.3, or 4.2.1 [27]. Specifically, we annotated single cells with plate metadata, which included treatment information. Next, we normalized each CellProfiler feature across all cells from the given plate using z-score normalization. We estimated the mean and standard deviation for the z-score transform using only cells from the prespecified negative control wells per plate.

The AreaShape CellProfiler features we used to train the phenotypic profiling model were not exactly the same as the CellProfiler features measured in the

CPJUMP1 pilot dataset. They differed by a single feature ("Nuclei_AreaShape_ConvexArea"). We set this missing feature in the CPJUMP1 dataset to zero to align feature spaces. This process leveraged the properties of our class-weighted multinomial logistic regression model to only compute probability estimates using measured features. After aligning the CellProfiler features, we generated the CPJUMP1 cell probabilities for each of the 15 MitoCheck phenotypes by applying the pre-trained class-weight balanced, AreaShape-only multi-class phenotypic profiling machine learning model (see machine learning training procedure methods section above). We also applied the model trained on shuffled input features as a negative control baseline.

Evaluating single-cell phenotype probability estimates in CPJUMP1

We compared the phenotype probabilities of each treated CPJUMP1 well to those of negative control cells on the same plate using KS tests. In other words, we tested for the difference in single-cell phenotype probability distributions in treatments versus controls within each CPJUMP1 plate for each phenotype. We chose KS tests because they are non-parametric and easily interpreted. We used the same CPJUMP1-defined negative control wells [27], of which differed based on the treatment type (DMSO = compound, non-targeting guides = CRISPR, lowly expressed genes = ORF). We removed treatment wells with fewer than 50 cells. To reliably compare treatment and negative control groups, we down-sampled the group with the highest cell count (majority group). When the majority group was the negative control group, we applied a stratified down-sample balanced by well to ensure the downsampled group had an equal representation across negative control wells. We applied the same procedure with probability estimates derived from models trained with randomly shuffled input data.

We aggregated single-cell phenotype probabilities per CPJUMP1 well using the median. This represents the central tendency of phenotype probabilities per well and is equivalent to aggregating single-cell morphology features to form well-level morphology profiles. We consider this aggregated measurement a "phenotypic profile".

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s44330-024-00014-3.

Supplementary Material 1.

Supplementary Material 2.

Acknowledgements

We would like to thank Dave Bunten, Erik Serrano, and Michael Lippincott for performing code review. We would also like to thank Jean-Karim Heriche and Thomas Walter for help interpreting and accessing the MitoCheck data.

Authors' contributions

Conceptualization: RK, GPW Data curation: JT, RK, CM, GPW Formal analysis: JT, RK, CM, GPW Funding acquisition: GPW Investigation: JT, RK, CM, GPW Methodology: JT, RK, CM, GPW Project administration: GPW Resources: GPW Software: JT, RK, CM, GPW Validation: JT, RK, CM, GPW Visualization: JT, RK, GPW Writing - original draft: JT, RK, CM, GPW Writing - review and editing: JT, RK, CM, GPW.

Funding

JT, CM, and GPW were supported in part by Alex's Lemonade Stand Foundation (Grant # 23–28306).

Data availability

We received the 2015 labeled dataset from the MitoCheck consortium and uploaded it for others to access and explore at https://github.com/WayScience/mitocheck_data [60]. Complete documentation for IDR Stream is available on GitHub at https://github.com/WayScience/IDR_stream. Code to perform all analyses (machine learning model training, validation experiments, figure making) is available at https://github.com/WayScience/phenotypic_profiling [61]. Code to process and evaluate the JUMP-CP pilot single-cell data is available at https://github.com/WayScience/JUMP-single-cell [62].

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Author details

[1]Department of Biomedical Informatics, University of Colorado School of Medicine, CO, USA. [2]Case Western Reserve University, OH, USA.

Received: 28 March 2024 Accepted: 16 October 2024
Published online: 09 December 2024

References

1. Ideker T, Galitski T, Hood L. A new approach to decoding life: systems biology. Annu Rev Genomics Hum Genet. 2001;2:343–72.
2. Tsherniak A, et al. Defining a Cancer Dependency Map. Cell. 2017;170:564-576.e16.
3. Caldera M, et al. Mapping the perturbome network of cellular perturbations. Nat Commun. 2019;10:5140.
4. Yin Z, et al. A screen for morphological complexity identifies regulators of switch-like transitions between discrete cell shapes. Nat Cell Biol. 2013;15:860–71.
5. Caicedo JC, et al. Data-analysis strategies for image-based cell profiling. Nat Methods. 2017;14:849–63.
6. Sero JE, et al. Cell shape and the microenvironment regulate nuclear translocation of NF-κB in breast epithelial and tumor cells. Mol Syst Biol. 2015;11:790.
7. Stirling DR, et al. Cell Profiler 4: improvements in speed, utility and usability. BMC Bioinformatics. 2021;22:433.
8. Moshkov, N. *et al.* Learning representations for image-based profiling of perturbations. *bioRxiv* 2022.08.12.503783 (2022) https://doi.org/10.1101/2022.08.12.503783.

9. Pfaendler, R., Hanimann, J., Lee, S. & Snijder, B. Self-supervised vision transformers accurately decode cellular state heterogeneity. bioRxiv. (2023) https://doi.org/10.1101/2023.01.16.524226.
10. Kraus O, et al. Masked Autoencoders are Scalable Learners of Cellular Morphology. arXiv. 2023. https://doi.org/10.48550/arXiv.2309.16064.
11. Caicedo JC, et al. Cell Painting predicts impact of lung cancer variants. Mol Biol Cell. 2022. https://doi.org/10.1091/mbc.E21-11-0538.
12. Way GP, et al. Predicting cell health phenotypes using image-based morphology profiling. Mol Biol Cell. 2021;32:995–1005.
13. Nyffeler J, et al. Bioactivity screening of environmental chemicals using imaging-based high-throughput phenotypic profiling. Toxicol Appl Pharmacol. 2020;389: 114876.
14. Schorpp K, et al. Cell DeathPred: a deep learning framework for ferroptosis and apoptosis prediction based on cell painting. Cell Death Discov. 2023;9:277.
15. Neumann B, et al. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. Nature. 2010;464:721–7.
16. Walter N, Eils R, Rohr K. Automated Classification of Mitotic Phenotypes of Human Cells Using Fluorescent Proteins. Methods Cell Biol. 2008;85:539–54 Academic Press.
17. Walter T, et al. Automatic identification and clustering of chromosome phenotypes in a genome wide RNAi screen by time-lapse imaging. J Struct Biol. 2010;170:1–9.
18. Pau G, et al. Dynamical modelling of phenotypes in a genome-wide RNAi live-cell imaging assay. BMC Bioinformatics. 2013;14:308.
19. Held M, et al. Cell Cognition: time-resolved phenotype annotation in high-throughput live cell imaging. Nat Methods. 2010;7:747–54.
20. Pratapa A, Doron M, Caicedo JC. Image-based cell phenotyping with deep learning. Curr Opin Chem Biol. 2021;65:9–17.
21. Dürr O, Sick B. Single-Cell Phenotype Classification Using Deep Convolutional Neural Networks. J Biomol Screen. 2016;21:998–1003.
22. Ulicna, K., Kelkar, M., Soelistyo, C. J., Charras, G. T. & Lowe, A. R. Learning dynamic image representations for self-supervised cell cycle annotation. *bioRxiv* (2023) https://doi.org/10.1101/2023.05.30.542796.
23. Celik, S. *et al.* Biological cartography: Building and benchmarking representations of life. *bioRxiv* (2022) https://doi.org/10.1101/2022.12.09.519400.
24. Way GP, et al. Morphology and gene expression profiling provide complementary information for mapping cell state. Cell Syst. 2022;13:911-923.e9.
25. Haghighi M, Caicedo JC, Cimini BA, Carpenter AE, Singh S. High-dimensional gene expression and morphology profiles of cells across 28,000 genetic and chemical perturbations. Nat Methods. 2022;19:1550–7.
26. Funk L, et al. The phenotypic landscape of essential human genes. Cell. 2022;185:4634-4653.e22.
27. Chandrasekaran, S. N. *et al.* JUMP Cell Painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *bioRxiv* (2023) https://doi.org/10.1101/2023.03.23.534023.
28. Jupp S, et al. The cellular microscopy phenotype ontology. J Biomed Semantics. 2016;7:28.
29. Williams E, et al. The Image Data Resource: A Bioimage Data Integration and Publication Platform. Nat Methods. 2017;14:775–81.
30. Peng T, et al. A BaSiC tool for background and shading correction of optical microscopy images. Nat Commun. 2017;8:14836.
31. Stringer C, Wang T, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. Nat Methods. 2021;18:100–6.
32. Serrano E, et al. Reproducible image-based profiling with Pycytominer. arXiv. 2023. https://doi.org/10.48550/arXiv.2311.13417.
33. McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software. 2018;3:861.
34. Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65.
35. van der Maaten L, Hinton G. Visualizing Data using t-SNE. J Mach Learn Res. 2008;9:2579–605.
36. Hirsch V, Reimann P, Treder-Tschechlov D, Schwarz H, Mitschang B. Exploiting domain knowledge to address class imbalance and a heterogeneous feature space in multi-class classification. VLDB J. 2023;32:1037–64.
37. Consoli U, et al. Differential induction of apoptosis by fludarabine monophosphate in leukemic B and normal T cells in chronic lymphocytic leukemia. Blood. 1998;91:1742–8.
38. Zhang X, et al. Purvalanol A induces apoptosis and reverses cisplatin resistance in ovarian cancer. Anticancer Drugs. 2023;34:29–43.
39. Qi Y, et al. Ribosomal protein L23 negatively regulates cellular apoptosis via the RPL23/Miz-1/c-Myc circuit in higher-risk myelodysplastic syndrome. Sci Rep. 2017;7:2323.
40. Gustafsdottir SM, et al. Multiplex cytological profiling assay to measure diverse cellular states. PLoS ONE. 2013;8: e80999.
41. Leung YY, Yao Hui LL, Kraus VB. Colchicine-Update on mechanisms of action and therapeutic uses. Semin Arthritis Rheum. 2015;45:341–50.
42. Zhao X, et al. Intracellular reduction in ATP levels contributes to CYT997-induced suppression of metastasis of head and neck squamous carcinoma. J Cell Mol Med. 2019;23:1174–82.
43. Cimini BA, et al. Optimizing the Cell Painting assay for image-based profiling. Nat Protoc. 2023;18:1981–2013.
44. Sypetkowski M, et al. RxRx1: A Dataset for Evaluating Experimental Batch Correction Methods. arXiv. 2023. https://doi.org/10.48550/arXiv.2301.05768.
45. Arevalo, J., van Dijk, R., Carpenter, A. E. & Singh, S. Evaluating batch correction methods for image-based cell profiling. *bioRxiv* (2023) https://doi.org/10.1101/2023.09.15.558001.
46. Tran HTN, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biol. 2020;21:12.
47. Ando, D. M., McLean, C. Y. & Berndl, M. Improving phenotypic measurements in high-content imaging screens. *bioRxiv* (2017) https://doi.org/10.1101/161422.
48. Lin, A. & Lu, A. X. Incorporating knowledge of plates in batch normalization improves generalization of deep learning for microscopy images. *bioRxiv* (2022) https://doi.org/10.1101/2022.10.14.512286.
49. Natekar, P., Wang, Z., Arora, M., Hakozaki, H. & Schöneberg, J. Self-supervised deep learning uncovers the semantic landscape of drug-induced latent mitochondrial phenotypes. *bioRxiv* (2023) https://doi.org/10.1101/2023.09.13.557636.
50. Seal S, et al. From pixels to phenotypes: Integrating image-based profiling with cell health data as BioMorph features improves interpretability. Mol Biol Cell. 2024;35(3):mr2.
51. Johnson GT, et al. Building the next generation of virtual cells to understand cellular biology. Biophys J. 2023;122:3560–9.
52. Viana MP, et al. Integrated intracellular organization and its variations in human iPS cells. Nature. 2023;613:345–54.
53. Sommer C, Held M, Fischer B, Huber W, Gerlich DW. Cell H5: a format for data exchange in high-content screening. Bioinformatics. 2013;29:1580–2.
54. Moore J, et al. OMERO and Bio-Formats 5: flexible access to large bioimaging datasets at scale. Proc SPIE Med Imaging. 2015;9413:37–42.
55. Rueden CT, et al. PyImageJ: A library for integrating ImageJ and Python. Nat Methods. 2022;19:1326–7.
56. Cooper GM. The Cell: A Molecular Approach. 2nd ed. Sunderland: Sinauer Associates; 2000. Available from: https://www.ncbi.nlm.nih.gov/books/NBK9839/.
57. Pedregosa F, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12:2825–30.
58. Gu Z, Eils R, Schlesner M. Complex heatmaps reveal patterns and correlations in multidimensional genomic data. Bioinformatics. 2016;32:2847–9.
59. Weisbart E, Kumar A, Arevalo J, et al. Cell Painting Gallery: an open resource for image-based profiling. Nat Methods. 2024;21:1775–7.
60. Kern, R. *WayScience/mitocheck_data: v3.0.0 - Manuscript Submission - Full MitoCheck Data Processing*. (Zenodo, 2024). https://doi.org/10.5281/ZENODO.10814990.
61. Way, G., Kern, R. & Tomkinson, J. *WayScience/phenotypic_profiling: v1.0 - Manuscript Submission*. (Zenodo, 2024). https://doi.org/10.5281/ZENODO.10814940.
62. Mattson, C., Way, G. & Tomkinson, J. *gwaybio/JUMP-Single-Cell: v1.0 - Manuscript Submission - JUMP-CP Pilot Data*. (Zenodo, 2024). https://doi.org/10.5281/ZENODO.10815000.

Publisher's Note