Contrastive learning for feature aggregation in image-based cell profiling

Robert van Dijk
Medical Image Analysis group
Eindhoven University of Technology

Abstract

Image-based cell profiling is a powerful tool that compares differently perturbed cell populations by measuring thousands of single-cell features and summarizing them into vectors (or profiles). Despite its simplicity, so-called average profiling, where all single-cell features are averaged using measures of center, is still the most commonly used approach. However, this method fails to capture cell populations' heterogeneity, which has been shown to improve the phenotypic strength of profiles. A recent study proposed a method that did capture cell population heterogeneity, but their method is difficult to use in practice. Therefore, we propose a Deep Sets based method that learns the best way of aggregating single-cell feature data into a profile that better predicts a compound's mechanism of action compared to average profiling. This is achieved by applying weakly supervised contrastive learning in a multiple instance learning setting. Our proposed model provides a more accessible and better performing method for aggregating single-cell feature data than previous studies. It is likely that the model achieves this by performing some form of quality control by filtering out noisy cells and prioritizing less noisy cells. Although it cannot be directly transferred to unseen experiment data, it could already be used by training on new data and inferring the improved profiles directly after, because the labels required for training are naturally available in cell profiling experiments. The application of this method could help improve the effectiveness of future cell profiling studies

Introduction

With the advent of high-throughput assays, quantification of cellular morphological responses at a large scale can be carried out. Image-based assays are among the most accessible and inexpensive technologies for this purpose. In these assays, cell populations are perturbed with compounds or genetic edits, dyed, and then imaged. By extracting large amounts of quantitative morphological data from these microscopy images, a profile can be created that describes that cell population's phenotype. The profiles of different cell populations can be compared to predict previously unrecognized cell states induced by different experimental perturbations of interest. This method, called image-based cell profiling, is a powerful tool that can be used for drug discovery, functional genomics, and disease phenotyping [1]. Among other things, cell profiling has already been used to find drugs for SARS-CoV-2 [2], work towards label-free leukemia detection [3], and predict the impacts of particular gene mutations [4].

Image-based cell profiling shows great potential, but many steps in its pipeline can still be improved [5]. One of the main challenges is to create a profile that summarizes the many features of a cell population while capturing their natural variations and subpopulations. Cell populations are known to be heterogeneous [6], [7] and recent studies have yielded many insights into its mechanisms and importance, particularly in cancer cells [8]–[11]. Capturing that heterogeneity could improve a profile's strength, i.e., its ability to find profiles created from similar cell populations.

Nevertheless, so-called average profiling, where all single-cell features are averaged using either the mean or the median, has remained the most commonly used approach in the field of image-based profiling, regardless of the type of features or the profile's postprocessing [12]. Average profiling is a simple way of summarizing a cell population (a sample) into a vector (a sample's profile) with only one value per measured feature. This step is required to decrease the data size (as there are typically thousands of cells per well, hundreds of wells per plate, and multiple plates per experiment) and to make downstream analysis more computationally viable.

However, by using average profiling, information on cell subpopulations is lost. This can result in identical average profiles despite cell populations having various subpopulation configurations. In that case, two profiles can be indistinguishable even though one is created from a sample that contains multiple subpopulations while the other sample does not contain any of those subpopulations. Additionally, not taking subpopulations into account can lead to a quantitatively incorrect interpretation. For example, two cell populations can show correlations among certain features when averaged but show completely different relations when compared after grouping the cells, i.e., Simpson's paradox [13]. Finally, by averaging a sample, the assumption that the joint distribution of the measured features is unimodal can lead to artifacts if violated.

Several methods have been proposed to capture the heterogeneity of cell populations into their corresponding profiles. The most straightforward solution is to incorporate the cell population's dispersion for each feature and concatenate these values with the average profile. However, this approach comes with its own limitations and only leads to minor improvements over average profiling alone [14], [15]. A different approach involves first clustering cells either in an unsupervised or supervised way and then calculating the profiles based on their subpopulations [16], [17]. These methods capture more information about subpopulations than only incorporating their dispersions; however, many cell phenotypes are better described with a continuous rather than a discrete scale [12]. Unsupervised clustering may reduce this discretization of cell phenotypes but can lead to incomparable profiles across samples due to a varying number of subpopulations. Moreover, these methods also did not significantly improve upon average profiling. In fact, a comparison study of profiling methods found that population means performed just as well as those that took advantage of cell heterogeneity [15].

Recently, however, the performance of average profiling was beaten by fusing features' averages, dispersion, and covariances [12]. This method strongly improved the performance in predicting a compound's mechanism of action (MoA) and a gene's pathway, showing that

capturing cell population heterogeneity can improve profile strength. However, this method requires users to clear multiple hurdles before it can be used. Thus, a more accessible method for capturing single-cell heterogeneity is required to increase profile strength in practice. That is why this study proposes a novel learning-based method that automatically finds the best way to aggregate single-cell data to improve the strength of sample profiles, *Figure 1a*.

One way to learn this aggregation is to apply a supervised approach, which uses manually or automatically annotated labels from individual cells based on their subpopulations. However, manual labeling is inherently not scalable, and automatic labeling encounters the same issues as mentioned above when clustering cell subpopulations [16], [17]. To overcome this challenge, a weakly-supervised contrastive learning approach is proposed that uses information naturally available in profiling experiments. Specifically, compound perturbation replicates are used as labels for learning a latent feature space. In this feature space, profiles of replicate perturbations should be close to each other and dissimilar profiles far away. This type of labeling frames the issue as a multiple-instance learning problem [18], which assumes that the replicate wells consist of cells with similar feature distributions. More specifically, each replicate well is assumed to contain at least one cell which is perturbed by the labeled compound.

The data is considered to be a collection of sets of cells, where each sample corresponds to one set. This requires a few properties from the function that aggregates these cells into a profile. Firstly, the function should be able to handle arbitrarily sized sets as an input. Secondly, because sets by definition have no order, the function should be permutation invariant. There are a few methods that have been developed for analyzing this type of data [19], [20], but a general formulation for solving this type of problem is known as Deep Sets [21]. Zaheer et al. show that a universal approximator on sets has a fixed form, which provides a backbone for building neural networks to process them. In this study, the Deep Sets architecture is used to learn the best way of aggregating single-cell feature data into a profile that allows for better prediction of a compound's MoA compared to the average profile. This is achieved by applying weakly supervised contrastive learning in a multiple instance learning setting. Improving the profile strength will increase the effectiveness of future cell profiling studies.

Methods

Our proposed model follows the general Deep Sets architecture [21]. The Deep Sets architecture can process permutation invariant and equivariant data and can learn to estimate first and second-order moments from the input data. Permutation invariance and equivariance are required for aggregating sets of single-cell data into a sample profile. Estimating the first and second-order moments is especially important as the previous study by Rohban et al. has shown that this can improve sample profile strength [12]. An experiment was performed to show that this is still the case in a weakly supervised contrastive learning setting, *Supplementary material A*.

After segmentation, thousands of features are measured for each cell, *Figure 1a*. These feature vectors are aggregated using a function $f(x)$ to get a single profile representing the cell population. There are multiple ways of defining the aggregation function $f(x)$. Our proposed architecture, *Figure 1b*, consists of two functions φ and ρ, which are simple fully connected neural networks, capable of approximating arbitrary polynomials. The model transforms the input set $\boldsymbol{X}$ by transforming each instance of the set $x_m \in \mathbb{R}^D$ by some neural network φ: $\mathbb{R}^D \rightarrow \mathbb{R}^N$. φ consists of a single fully connected layer with 2048 nodes followed by a leaky ReLU activation layer. All of these nonlinear representations φ(x_m) are summed, collapsing the cell dimension M. The output $z \in \mathbb{R}^N$ is then processed by the projection network ρ: $\mathbb{R}^N \rightarrow \mathbb{R}^L$, which applies more nonlinear transformations to create a final representation in the loss space. The function ρ consists of two subsequent fully connected layers of 512 and 2048 nodes respectively, both followed by leaky ReLU activations.

The model is trained using contrastive learning by computing the Supervised Contrastive (SupCon) loss [22]. This loss is different from the commonly used SimCLR [23] in only one aspect: it takes into account all positive examples of a certain sample instead of only one, *equation 1*. This SupCon loss pulls replicate samples, or positive pairs, together in the embedding space, while simultaneously pushing apart samples perturbed with different compounds, or negative pairs, *Figure 1c*. In this study, cosine similarity is used as the distance metric. The loss shows benefits for robustness to natural corruptions, hyperparameter settings, and inherently performs hard positive and hard negative mining when used in combination with cosine similarity [22].

$$L^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(z_i \bullet z_p / \tau)}{\sum_{a \in A(i)} exp(z_i \bullet z_a / \tau)} \quad (1)$$

I: total number of samples
$P(i)$: number of positive samples for the current sample i
$A(i)$: number of negative samples for the current sample i
τ: temperature constant (hyperparameter)

After training the model, the projection network ρ is often discarded, and the summed representation z is used for downstream analysis instead [21]. However, ρ is not discarded here because the projection it has learned is tied to the evaluation task. One of the main applications of image-based cell profiling is discovering the unknown MoA of a certain compound. To that end, in addition to replicate prediction (the training task), the proposed model will be evaluated using MoA prediction. MoA prediction is evaluated by quantifying a profile's ability to find its so-called sister compound, which is a different compound with the same annotated MoA. If the model has learned to amplify the phenotypic signature of a sample's profile, finding its sister compound should also become easier.

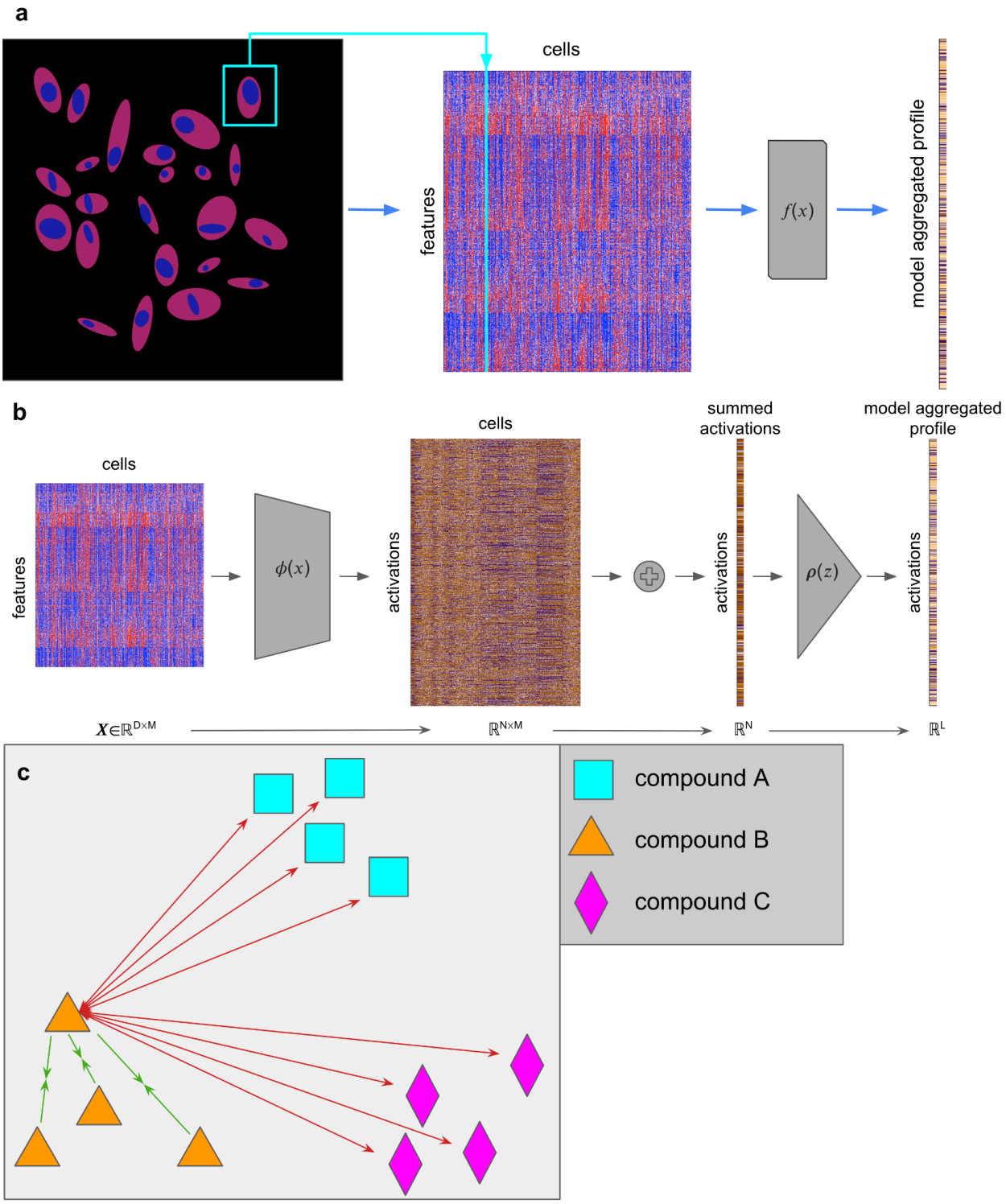


Figure 1: (a) Thousands of features are extracted from each segmented cell in microscopy images of wells. A learned function $f(x)$ aggregates this data into a single feature vector: the sample's profile. (b) An in-depth look at the model architecture used in this study. The model consists of three elements: a function $\varphi(x)$, which maps the input data from $\mathbb{R}^D$ to $\mathbb{R}^N$ space, a summation, which collapses the cell dimension, and $\rho(z)$, which maps the collapsed representation from $\mathbb{R}^N$ to $\mathbb{R}^L$ space. (c) During training, replicate compound profiles are forced to attract each other (green arrows) and simultaneously repel every other compound (red arrows) in the learned feature space. Here, all forces are drawn for a single profile of compound B.

The performance in MoA prediction and replicate prediction will be compared between model-based profiling, which uses the learned aggregation, and average profiling. The performance is quantified by calculating the mean average precision (mAP). Average precision (AP) is an information retrieval metric that indicates whether the model can correctly identify all the positive examples without accidentally marking too many negative examples as positive. Specifically, the metric is equal to the area under the precision-recall curve. The calculation for the AP of a single query is shown in *equation 2*. To compute the AP, a rank order of sample profiles is required. The top of the rank order corresponds to profiles most similar to the queried profile and vice versa. This rank order is created by calculating the cosine similarity between all K sample profiles.

$$AP = \sum_{k=1}^{K} (r(k) - r(k-1))p(k) \quad (2)$$

K: total number of sample profiles
$p(k)$: precision at the k-th position of the rank order
$r(k)$: recall at the k-th position of the rank order

A compound's AP is calculated by taking the average AP over all its replicate compound profiles, while a MoA's AP additionally averages over sister compound profiles. The mAP is the mean AP over all of the compounds in a plate. If a compound does not have an annotated sister compound, it and its replicates are left out of the mAP computation for MoA prediction.

Deep learning models are notoriously difficult to interpret. However, interrogating them has led to useful new insights in previous studies [24] and allows users to better understand the reasoning behind a model's output. In this study, a combination of sensitivity analysis (SA) [25] and a method similar to critical point analysis (CPA) used in the PointNet study [26] was used to investigate possible biological foundations for the model's output.

SA explains a model's prediction based on locally evaluated partial derivatives. The partial derivatives of a model's output $f(\boldsymbol{X})$ are calculated with respect to each entry in the input matrix $x_{m,d}$ by backpropagating the loss function L^{sup}. Afterward, the absolute value is taken from these partial derivatives, *equation 3*. These values can then be summed over either the cell or feature dimensions to get the feature or cell relevance scores, respectively. In this case, the relevance score per cell is computed. The analysis assumes that the most relevant input values are those to which the model's output is the most sensitive. Thus, inputs that receive a high relevance score will, when changed, make it more or less likely for the model to make a certain prediction. As a result, high relevance values can also characterize input patterns that the model would like to see removed to improve its performance for the predicted class. These patterns, e.g., noise, may not be linked to the class of interest.

$$R_{m,d} = \left|\left|\frac{\delta}{\delta x_{m,d}} f(\boldsymbol{X})\right|\right| \quad (3)$$

To counteract some of the potential noisy predictions of SA, CPA is used additionally for calculating the relevance scores. Since the model architecture is permutation invariant, each input cell vector is processed independently. In the PointNet study [26], CPA consisted of finding the input points with the maximum value for each feature. These points were found to form the skeleton of the input, meaning that they are the most relevant points for defining it. Their model's permutation invariant operation was a max pooling operation that inherently selected these points. In this study, a summation is used instead, although the reasoning is the same: cells with high activation values before the permutation invariant operation contribute more to the output of the model than those with low activation values. The CPA relevance score was calculated per cell by taking the L1-norm of their respective activations of the first fully connected layer.

The relevance scores of SA and CPA are min-max normalized per well and then combined. The combination of the two is again min-max normalized, resulting in the combined relevance score. This score was used instead of the separate relevance scores because averaging was expected to cancel out some of the potentially noisy predictions. In fact, the combined relevance score was found to have higher Pearson correlations with the CellProfiler features than the separate relevance scores, *Supplementary material D*. Cells with the highest (>0.8) or lowest (<0.2) combined relevance scores will hereon be denoted as the 'most relevant' or 'least relevant' cells.

Multiple methods are used to investigate what the model has learned. First, the model and average aggregated profiles of the top 15 sister compound pairs, based on their mAP for MoA prediction, are visualized using a UMAP [27]. Four different values were tried for the UMAP hyperparameter $n\_neighbors$, which balances local versus global structure in the data. From these UMAPs, one was chosen based on the best presentation of the clusters that were consistently visible throughout all UMAPs. The second method calculates the Pearson correlation between all of the input CellProfiler features and the combined relevance scores to get a better understanding of what features the model prioritizes. Finally, the most and least relevant cells are visualized and analyzed in the raw microscopy images to link the model's output to the underlying cell biology of the compound perturbations. Plates from multiple datasets are used for these analyses to find commonalities between them.

Experimental setup

The cpg0000 dataset [28], from the JUMP consortium [29], was used, available from the Cell Painting Gallery on the Registry of Open Data on AWS (https://registry.opendata.aws/cellpainting-gallery/). Specifically, single-cell level data from four experiments called Stain2, Stain3, Stain4, and Stain5 were used. These datasets were created with the aim of optimizing the analysis pipeline for image-based cell profiling. This means that multiple batches with a different analysis pipeline exist within each dataset. The analysis pipeline varied in dye concentration, cell permeabilization, cell seeding, exposure, pixel binning, compound dose, or microscopy method (confocal versus widefield). Each dataset (Stain2,

Stain3, Stain4, or Stain5) aimed to optimize the profile signal strength by adjusting one or multiple of these factors.

These datasets used 384-well plates. Each well was seeded with U2OS cells (a bone cancer cell line) and perturbed with one of 90 compounds. The well position of each compound perturbation was fixed, i.e., the same plate layout was used, across all datasets. Each compound was replicated four times per plate. The other 24 wells were used as negative controls, which contained cells in the solvent dimethyl sulfoxide (DMSO) only. The cells were stained using the Cell Painting protocol [30], after which images were taken with a microscope. From each well, either four (Stain2) or nine (Stain3, Stain4, and Stain5) images were taken with different field-of-views (FOVs).

After the cells were segmented in each image, features were extracted using CellProfiler [31]. Per cell 4295 features were extracted from Stain2, Stain3, and Stain4, and 5794 features were extracted from Stain5 due to an update to the feature extraction pipeline. Because the Deep Sets model requires a fixed number of input features and to reduce the model's computational burden, a subset of 1324 features was taken which were available in all four datasets. This made the model easily transferable between the four different Stain datasets. The complete list of these feature names can be found in the GitHub repository of this project [32]. The features were standardized on the plate level to reduce batch effects, resulting in zero mean and unit variance features across all cells in the plate. The negative control wells were then removed from all plates because by definition these do not have a strong profile.

A commonly used pipeline was employed to calculate the average profiles. Only the selected 1324 features were used to allow for a fair comparison with the model aggregated profiles. After calculating the average profile for each well in a certain plate, the features were RobustMAD normalized by subtracting their median and dividing by their mean absolute deviation. The final average profiles were acquired by applying feature selection using a variance and correlation threshold. For the model, the outputs were used directly as the aggregated profiles.

The data was stratified in two ways. First, the 90 compounds were split into 72 (or 80%) training and 18 (or 20%) validation compounds. These were the same across all plates. Second, all available plates were stratified per dataset, except for Stain5, resulting in 5 training, 4 validation, and 3 test plates for Stain2, Stain3, and Stain4. Six plates from Stain5 were used as an out-of-distribution test set with different experimental conditions than the training data. *Supplementary material B* shows the plate names per dataset for this stratification. These stratifications allowed for isolated evaluation of the model's ability to generalize to unseen compound distributions and unseen plate distributions.

Unfortunately, batch effects, which describe the technical differences between various plates, can be quite severe in these datasets due to differences in their analysis pipeline. Technical differences between the different StainX experiments (or experiment effects) can be even larger. Since the aim of this study is not to solve the batch or experiment effect problem, the model is not expected to perform well on the out-of-distribution test set Stain5. However, it is important to

quantify the limitations of the model. The test set plates chosen for Stain2, Stain3, and Stain4 are considered out-of-distribution plates due to batch effects. This choice was based on how similar they are to the training and validation data. *Supplementary material E* describes how this similarity is measured.

Because there was a limited number of compounds available, data augmentation was used to increase the generalization of the model to unseen compounds. During training, for each batch, a number of cells was randomly sampled with replacement from every well. The number of cells that was sampled was itself a number sampled from a Gaussian distribution with a mean and standard deviation of 2000 and 900, respectively. Although they remained mostly similar, this created unique sets of cells for each compound in every batch. Four sets of cells were sampled for each compound in each epoch. This number was chosen because there are four replicates of each compound on a plate. However, each sampled set could consist of cells from a single well or a combination of two at random. This decision was based on a coin flip, which was performed for each sample. This last form of augmentation should help decrease plate-layout effects, which introduce a technical bias into the single-cell feature data based on which well position the population is in.

The model training process consists of many tunable hyperparameters, which were chosen using a random search. The AdamW optimizer [33] is used with a learning rate of 5×10^{-4} and weight decay 10^{-2}. The model is trained for 100 epochs or until the best mAP is achieved on the validation compounds of the training plates. A batch size of 72 is used, consisting of four samples of 18 different compounds. An overview of all the tunable parameters, including the data augmentation parameters, is given in *Supplementary material C*.

Results

The model's aggregated profiles consistently show significantly higher mAP scores for replicate training and validation compound prediction than the average profiles for the training and validation plates of Stain2, Stain3, and Stain4, *Figure 2*. The increase in training compound mAP by using model aggregation is generally higher than that for validation compounds. On the test plates, the model aggregation does not achieve significantly higher mAP scores than average profiling, except for the training compounds of Stain4. However, the mAP scores using model aggregation for the test plates of Stain2, Stain3, and Stain4 are not lower than those achieved with average profiling. The mAP scores for Stain5 are either similar to or lower than those achieved with average profiling.

The model's aggregated profiles consistently show higher mAP scores for MoA prediction than the average profiles for the training, validation, and test plates of Stain2, Stain3, and Stain4, *Figure 3*. The model aggregation only slightly improves the mAP upon average profiling for two plates in the test set of Stain2. As a result, the test set of Stain2 shows the only non-significant difference between the mean of the model aggregated and average profile mAP scores

(p=0.17). The average profiles generally achieve a higher mAP than the model aggregated profiles on the Stain5 test set. The model aggregated profiles, on average, achieve a mAP that is 68%, 61%, and 49% higher than average profiling for the training, validation, and test set of Stain2, Stain3, and Stain4, *Table 1*. Average profiling achieves a mAP that is 36% higher on average than the model aggregated profile for Stain5.

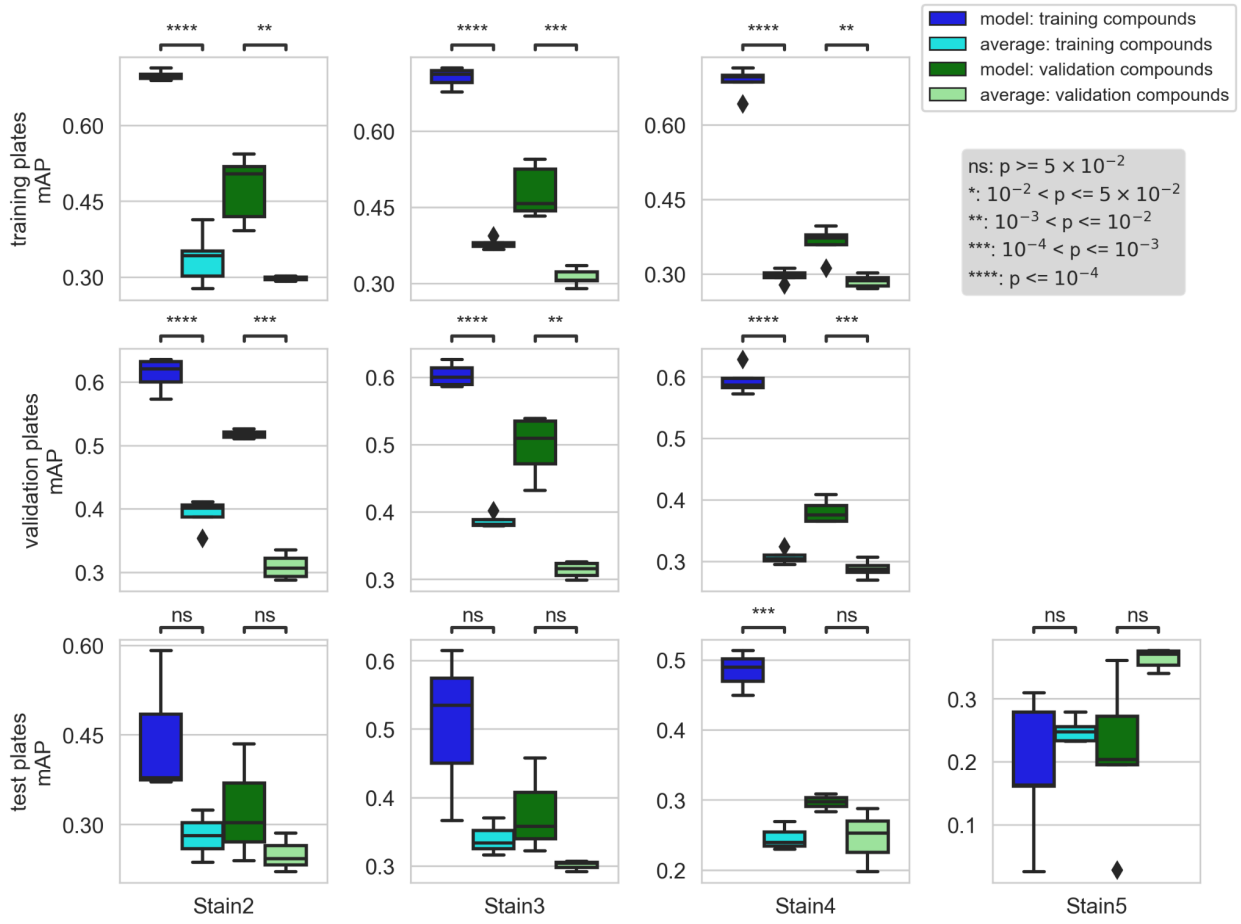


Figure 2: mAP boxplots of replicate prediction for training and validation compounds by the model, dark blue and cyan respectively, and average, dark green and light green respectively, aggregated profiles. mAP scores were calculated separately per training, validation, and test set (rows) and per experiment dataset Stain2, Stain3, Stain4, and Stain5 (columns). Welch's t-tests were used to compare the means between the model and average mAP scores on corresponding data; their p-values are indicated as stars at the top of each plot.

The UMAP of the mean aggregated profiles shows three clusters of sister compound profiles that are easily distinguishable from the rest of the profiles, *Figure 4a*. The other compound profiles are either part of the main cluster in the center of the figure or one of the three smaller clusters in the top left, bottom, or far-right of the figure. The far-right cluster contains profiles from only a single plate. At the edges of the main cluster, three more clusters of single plates are identified. No well clusters are identified outside of the sister compound clusters.

The UMAP of the model aggregated profiles shows six clusters of sister compound profiles that are easily distinguishable from the rest of the profiles, *Figure 4b*. All other clusters are also grouped by compound but can be found either in isolation or in the main cluster in the center of the figure. No clusters are formed according to plates or wells, i.e., other than the sister compound profiles.

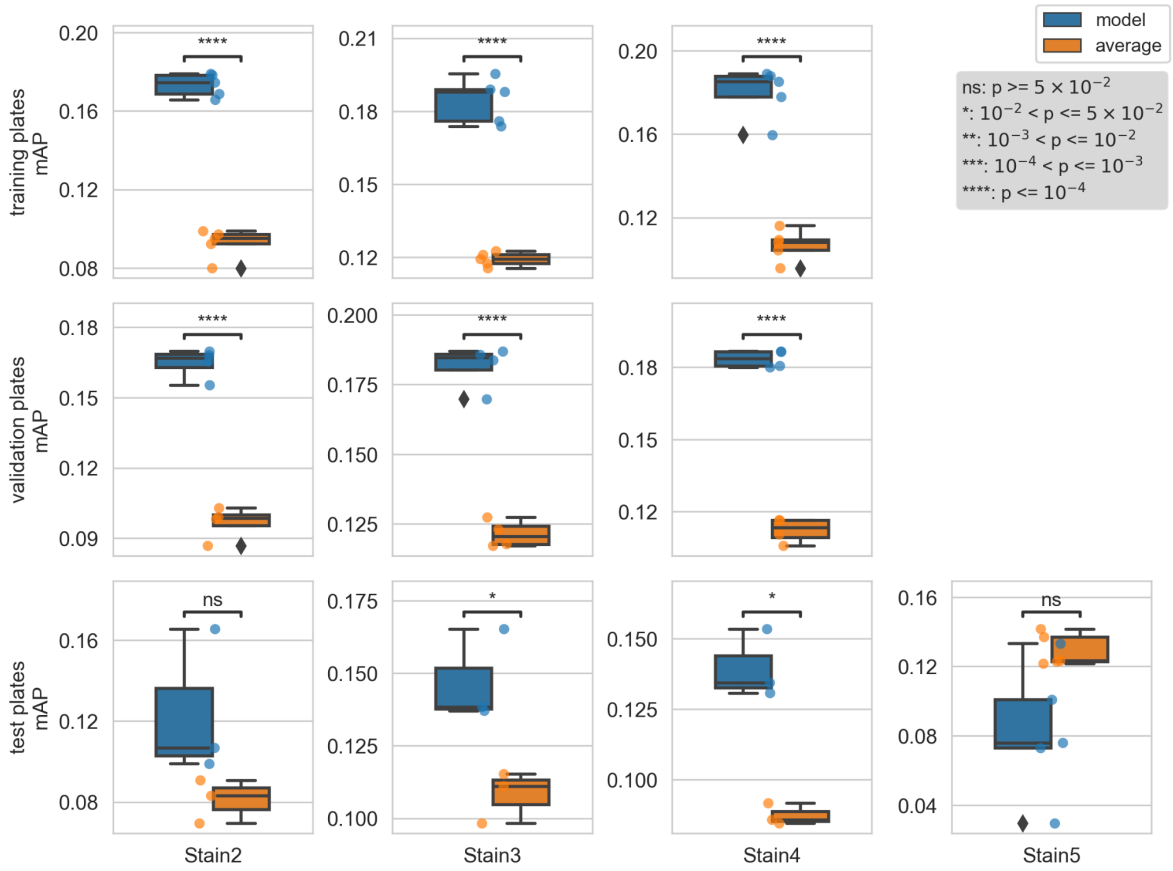


Figure 3: mAP boxplots of MoA prediction for model (blue) and average (orange) aggregated profiles. mAP scores were calculated separately per training, validation, and test set (rows) and per experiment dataset Stain2, Stain3, Stain4, and Stain5 (columns). Welch's t-tests were used to compare the means between the model and average mAP scores; their p-values are indicated as stars inside each plot.

Table 1: Absolute and relative average differences in mAP of MoA prediction between model and average aggregated profiles. The differences are calculated as $mAP(model) - mAP(average)$.

Stratification	mAP difference Stain2	mAP difference Stain3	mAP difference Stain4	average mAP difference Stain2, Stain3, and Stain4	mAP difference Stain5
Training	0.081 (81%)	0.065 (55%)	0.073 (69%)	0.073 (68%)	
Validation	0.068 (70%)	0.060 (50%)	0.071 (63%)	0.066 (61%)	
Test	0.043 (52%)	0.039 (36%)	0.052 (60%)	0.045 (49%)	-0.047 (-36%)

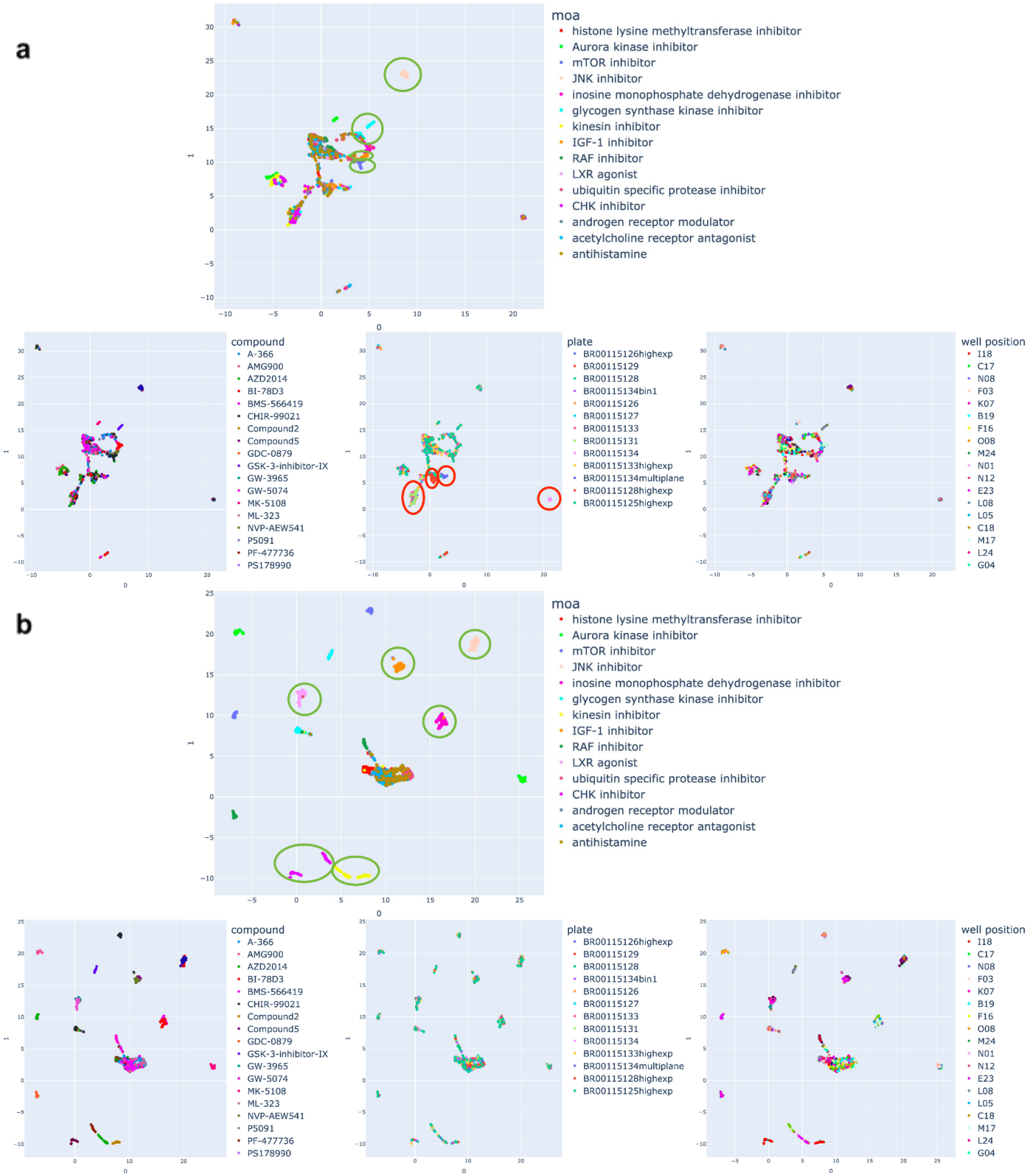


Figure 4: UMAP of the mean (a) and model (b) aggregated profiles of the top 15 MoAs, based on the model's mAP scores for MoA prediction, from all used Stain3 plates. The UMAP was created using $n\_neighbors = 15$ and cosine similarity as a distance measure. The profiles are colored based on their corresponding annotated moa (top), compound (bottom left), plate (bottom middle), and well position (bottom right). Clusters of sister compound profiles that were visible for multiple $n\_neighbors$ values are annotated in green and isolated plate clusters are annotated in red.

Table 2: Top 5 CellProfiler features based on their positive and negative Pearson correlation coefficient with the SA and CPA combined relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	Feature name	Correlation coefficient
Cytoplasm	Cytoplasm_Correlation_K_DNA_Brightfield	0.72
Cells	Cells_AreaShape_MeanRadius	0.71
Cells	Cells_AreaShape_MaximumRadius	0.70
Cells	Cells_AreaShape_MedianRadius	0.70
Cells	Cells_AreaShape_Area	0.68

Feature category	Feature name	Correlation coefficient
Cells	Cells_Intensity_MeanIntensityEdge_DNA	-0.74
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_DNA	-0.72
Cytoplasm	Cytoplasm_Intensity_UpperQuartileIntensity_DNA	-0.71
Cytoplasm	Cytoplasm_Intensity_MeanIntensity_DNA	-0.69
Cytoplasm	Cytoplasm_Correlation_K_Brightfield_DNA	-0.67

The CellProfiler features that are the most highly correlated with the SA and CPA combined relevance score are either cytoplasm or general cell related features, as opposed to nuclei or image related features, *Table 2*. Features associated with AreaShape, i.e., with respect to the size and shape of the cell, and correlation between the DNA and brightfield channel images, which is a measure of cell crowdedness, are the most positively correlated with the combined relevance score (correlation coefficient of 0.68 to 0.72). On the other hand, features associated with DNA intensity and correlation between the brightfield and DNA channel images are the most negatively correlated with the combined relevance score (correlation coefficient of -0.67 to -0.74). *Supplementary material D* lists an additional 15 most correlated features.

The most relevant cells are generally isolated from other cells and do not contain spots of high-intensity pixels (green arrow), *Figure 5*. The least relevant cells exhibit the opposite behavior; they are more often clumped together (bottom red arrow) or contain spots of high-intensity pixels (top left and top right red arrows). The other three FOVs from plate BR00112197binned are shown in *Supplementary Material F*.

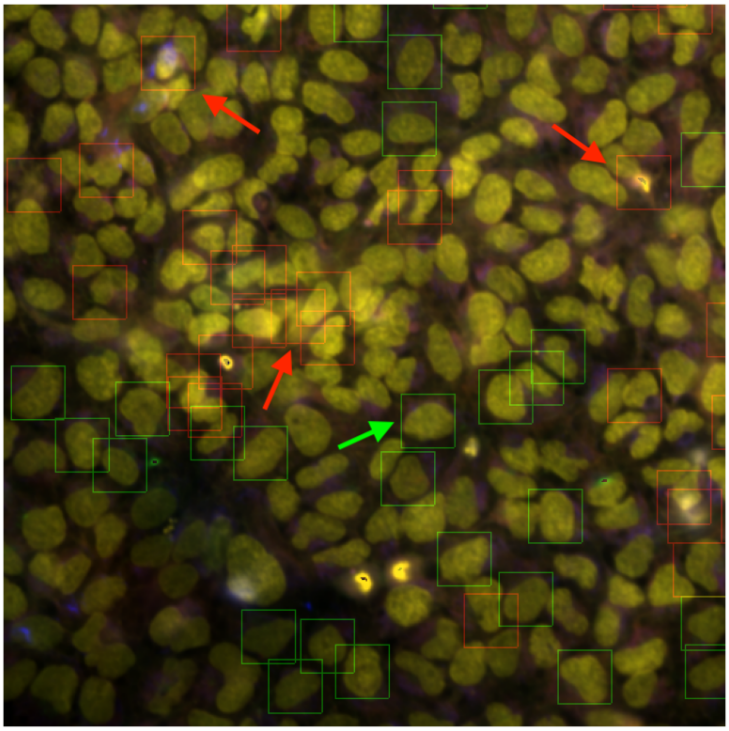


Figure 5: 5-channel combined microscope image of one of the four FOVs for plate BR00112197binned. The most relevant cells are annotated with green boxes and the least relevant cells are annotated with red boxes. Three cells characteristic for low relevance scores are explicitly labeled with red arrows. One cell characteristic for high relevance scores is explicitly labeled with a green arrow.

Discussion

Rohban et al. have shown that capturing heterogeneity in a cell population's profile can improve the profile's ability to predict a compound's MoA [12]. However, their proposed method is hard to use in practice. Therefore this study proposes a Deep Sets based model that automatically finds the best way to aggregate cell populations by using weakly supervised contrastive learning. This method solves the problem end-to-end by receiving single-cell feature input data and providing the aggregated profile as an output. The model achieves 36% to 60% better performance in MoA prediction than average profiling on hold-out data with strong batch effects and 50% to 81% better performance on data created with more similar conditions. Not only is this increase much higher than that of Rohban et al., who reported an increase of at least ~20%, it also provides a more accessible method which requires less input from the user.

The model has successfully learned a general method of aggregating single cell feature data so that seen and unseen compound profiles are more likely to find their replicates. Although the improvement that model aggregation provides for validation compounds is smaller than that for training compounds, it is significant with respect to average profiling for the training and validation plates. However, the model has more trouble with generalizing to out-of-distribution plates; it fails to significantly improve upon average profiling on the test set of most datasets for

both the training and validation compounds. It is possible that adding more data points would lead to a significant difference, as the model boxplots show generally higher mAP scores than the average boxplots.

Although the model was only trained to find replicate compounds, its ability to find sister compounds significantly improves upon average profiling. This suggests that the model learned to aggregate single-cell feature data in a way that preserves phenotypic information, which describes the underlying biological processes. However, it cannot do this for all test set plates. On two test plates in Stain2 the model achieved only a slight improvement in mAP. One of these two plates (BR00112199) was imaged using multiplane microscopy, which is a completely different image acquisition method than brightfield microscopy, which is used for all other plates. The other plate (BR00113821) had a lower cell seeding of 1250 instead of the usual 2500. These differences in the analysis pipeline could explain the relatively lower mAP achieved by the model on these plates.

As expected, the model could not beat average profiling in MoA or replicate prediction when it was used on a dataset created under different experimental conditions (Stain5). The model achieved similar performance to average profiling for only one plate (BR00120526), which had the same analysis pipeline as in some Stain4 plates, and worse performance for the rest overall. The aim of this study was not to solve the batch effect problem, and different experimental conditions can cause even larger technical variations in the data. The model can still be used on out-of-distribution plates as long as they were generated within the same experiment as the training data.

The model creates a better-organized feature space for discerning different compounds than average profiling. Compounds with strong phenotypic signatures are easily distinguishable in the UMAP, while compounds with weaker phenotypic signatures are clustered together in the center. Even though most compounds are clustered tightly, not all of them are clustered next to their sister compound. It is possible that specifically the latter compounds have off-target effects or different MoAs than the ones that are annotated. Another explanation could be the influence of well position effects, which introduce a bias. As opposed to model-based profiling, average profiling results in clustering of most compound profiles near the center of the UMAP, making them harder to distinguish. These profiles are also strongly influenced by batch effects, creating separate plate clusters. Remarkably, model-based profiling can cancel out these batch effects almost entirely, even though this was not part of the training objective.

The combined relevance scores in combination with the annotated microscope image provide useful insights into which cells the model may be prioritizing during aggregation. In the image, the most salient cells are generally isolated from other cells, while the least salient cells appear to be more confluent and tend to contain spots of high intensity pixels. The spots with locally high intensities can be explained in two ways. Some of these intensities are most likely artifacts or debris because they have sharp pixel intensity drop-offs and pointy shapes, e.g., the cell annotated with a red arrow in the top right of the image. The other intensities are most likely cells in some stage of cell division because they are small, bright, and sometimes have a twin next to it, e.g., the cell annotated with a red arrow in the top left of the image.

The correlation between the CellProfiler features and the combined relevance score affirms these findings. The combined relevance score is positively correlated with features that describe the size of the cell, which in turn is correlated with isolation as that allows these cells to expand. On the other hand, negatively correlated features generally describe the pixel intensity of DNA. The latter can be associated with cells that have just replicated their DNA, artifacts, or debris. All things considered, the model appears to have learned to identify features that describe the quality of the data. The model prioritizes high-quality cells, which have room to expand and can express the phenotypic effects of a compound in the best way, over low-quality cells, which may contain bias in its measured features due to cell division, artifacts, or debris.

Using the proposed model as an aggregation method for single-cell feature data can significantly improve MoA prediction performance compared to average profiling. However, the model's performance is limited to data that was created under similar experimental conditions as the training data. Although not ideal, this will likely not be an issue in practice. In most profiling experiments, plates are created using the same analysis pipeline, which means their features are more consistent and thus like the training and validation sets presented in this paper. In fact, a hold-out compound set may also not be required. The labels required for training the model are always available in profiling experiments. Thus, the model could also be trained using the compound replicates and then be used to infer the improved profiles for MoA prediction afterwards.

Although this last method could already be applied in practice, future research will need to find out how robust the current training setup is with respect to different datasets. One way to test this is to train the model on a larger dataset with limited technical variations using the same training setup and then evaluating if the model can still improve upon average profiling. This study has shown that the model has learned a sort of cell quality control filter, which allows it to improve profile strength. Although supporting experiments have shown that the model is able to learn second-order interactions, *Supplementary material A*, proving that it is doing so is a challenging task which requires its own line of research. Another future direction would be to extract a general set of single-cell feature aggregation rules from the model that can be used to improve profile strength and potentially refine our understanding of cell population heterogeneity.

Conclusion

Our proposed model provides a more accessible and better performing method for aggregating single-cell feature data than previous studies. It is likely that the model achieves this by performing some form of quality control by filtering out noisy cells and prioritizing less noisy cells. Remarkably, the model could also cancel out batch effects, even though this was not part of the training objective. Although it cannot be directly transferred to unseen experiment data, it could already be used by training on new data and inferring the improved profiles directly after because the labels required for training are naturally available in cell profiling experiments. The application of this method could help improve the effectiveness of future cell profiling studies.

References

[1] J. C. Caicedo, S. Singh, and A. E. Carpenter, “Applications in image-based profiling of perturbations,” *Curr. Opin. Biotechnol.*, vol. 39, pp. 134–142, Jun. 2016.

[2] C. Mirabelli *et al.*, “Morphological cell profiling of SARS-CoV-2 infection identifies drug repurposing candidates for COVID-19,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 118, no. 36, Sep. 2021, doi: 10.1073/pnas.2105815118.

[3] M. Doan *et al.*, “Label-Free Leukemia Monitoring by Computer Vision,” *Cytometry A*, vol. 97, no. 4, pp. 407–414, Apr. 2020.

[4] J. C. Caicedo *et al.*, “Cell Painting predicts impact of lung cancer variants,” *Mol. Biol. Cell*, vol. 33, no. 6, p. ar49, May 2022.

[5] S. N. Chandrasekaran, H. Ceulemans, J. D. Boyd, and A. E. Carpenter, “Image-based profiling for drug discovery: due for a machine-learning upgrade?,” *Nat. Rev. Drug Discov.*, vol. 20, no. 2, pp. 145–159, Feb. 2021.

[6] S. J. Altschuler and L. F. Wu, “Cellular heterogeneity: do differences make a difference?,” *Cell*, vol. 141, no. 4, pp. 559–563, May 2010.

[7] K. A. Janes, “Single-cell states versus single-cell atlases - two classes of heterogeneity that differ in meaning and method,” *Curr. Opin. Biotechnol.*, vol. 39, pp. 120–125, Jun. 2016.

[8] A. Marusyk and K. Polyak, “Tumor heterogeneity: causes and consequences,” *Biochim. Biophys. Acta*, vol. 1805, no. 1, pp. 105–117, Jan. 2010.

[9] D. Deb *et al.*, “Combination Therapy Targeting BCL6 and Phospho-STAT3 Defeats Intratumor Heterogeneity in a Subset of Non-Small Cell Lung Cancers,” *Cancer Res.*, vol. 77, no. 11, pp. 3070–3081, Jun. 2017.

[10] L. Keller and K. Pantel, “Unravelling tumour heterogeneity by single-cell profiling of circulating tumour cells,” *Nat. Rev. Cancer*, vol. 19, no. 10, pp. 553–567, Oct. 2019.

[11] J. Goveia *et al.*, “An Integrated Gene Expression Landscape Profiling Approach to Identify Lung Tumor Endothelial Cell Heterogeneity and Angiogenic Candidates,” *Cancer Cell*, vol. 37, no. 1, pp. 21–36.e13, Jan. 2020.

[12] M. H. Rohban, H. S. Abbasi, S. Singh, and A. E. Carpenter, “Capturing single-cell heterogeneity via data fusion improves image-based profiling,” *Nat. Commun.*, vol. 10, no. 1, p. 2082, May 2019.

[13] C. Trapnell, “Defining cell types and states with single-cell genomics,” *Genome Res.*, vol. 25, no. 10, pp. 1491–1498, Oct. 2015.

[14] B. Wang *et al.*, “Similarity network fusion for aggregating data types on a genomic scale,” *Nat. Methods*, vol. 11, no. 3, pp. 333–337, Mar. 2014.

[15] V. Ljosa *et al.*, “Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment,” *J. Biomol. Screen.*, vol. 18, no. 10, pp. 1321–1329, Dec. 2013.

[16] L.-H. Loo, H.-J. Lin, R. J. Steininger 3rd, Y. Wang, L. F. Wu, and S. J. Altschuler, “An approach for extensibly profiling the molecular states of cellular subpopulations,” *Nat. Methods*, vol. 6, no. 10, pp. 759–765, Oct. 2009.

[17] F. Fuchs *et al.*, “Clustering phenotype populations by genome-wide RNAi and multiparametric imaging,” *Mol. Syst. Biol.*, vol. 6, p. 370, Jun. 2010.

[18] Maron and Lozano-Pérez, “A Framework for Multiple-Instance Learning,” *Adv. Neural Inf. Process. Syst.*, Jun. 1997, [Online]. Available: https://proceedings.neurips.cc/paper/1997/file/82965d4ed8150294d4330ace00821d77-Paper.pdf

[19] H. Edwards and A. Storkey, “Towards a Neural Statistician,” *arXiv [stat.ML]*, Jun. 07, 2016.

[Online]. Available: http://arxiv.org/abs/1606.02185
[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” *arXiv [cs.CV]*, Jun. 07, 2017. [Online]. Available: http://arxiv.org/abs/1706.02413
[21] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, “Deep Sets,” *arXiv [cs.LG]*, Mar. 10, 2017. [Online]. Available: http://arxiv.org/abs/1703.06114
[22] P. Khosla *et al.*, “Supervised Contrastive Learning,” *arXiv [cs.LG]*, Apr. 23, 2020. [Online]. Available: http://arxiv.org/abs/2004.11362
[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” *arXiv [cs.LG]*, Feb. 13, 2020. [Online]. Available: http://arxiv.org/abs/2002.05709
[24] S. Chakraborty *et al.*, “Interpretability of deep learning models: A survey of results,” in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–6.
[25] W. Samek, T. Wiegand, and K.-R. Müller, “Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models,” *arXiv [cs.AI]*, Aug. 28, 2017. [Online]. Available: http://arxiv.org/abs/1708.08296
[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” *arXiv [cs.CV]*, Dec. 02, 2016. [Online]. Available: http://arxiv.org/abs/1612.00593
[27] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” *arXiv [stat.ML]*, Feb. 09, 2018. [Online]. Available: http://arxiv.org/abs/1802.03426
[28] S. N. Chandrasekaran *et al.*, “Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations,” *bioRxiv*, p. 2022.01.05.475090, Jan. 05, 2022. doi: 10.1101/2022.01.05.475090.
[29] A. Mullard, “Machine learning brings cell imaging promises into focus,” *Nat. Rev. Drug Discov.*, vol. 18, no. 9, pp. 653–655, Sep. 2019.
[30] M.-A. Bray *et al.*, “Cell Painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes,” *Nat. Protoc.*, vol. 11, no. 9, pp. 1757–1774, Sep. 2016.
[31] D. R. Stirling, M. J. Swain-Bowden, A. M. Lucas, A. E. Carpenter, B. A. Cimini, and A. Goodman, “CellProfiler 4: improvements in speed, utility and usability,” *BMC Bioinformatics*, vol. 22, no. 1, p. 433, Sep. 2021.
[32] R. van Dijk, *Contrastive learning based point set aggregation for image-based cell profiling*. 6 2022. [Online]. Available: https://github.com/broadinstitute/FeatureAggregation_single_cell
[33] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” Sep. 27, 2018. Accessed: Jun. 20, 2022. [Online]. Available: https://openreview.net/pdf?id=Bkg6RiCqY7
[34] R. Vemulapalli and D. W. Jacobs, “Riemannian Metric Learning for Symmetric Positive Definite Matrices,” *arXiv [cs.CV]*, Jan. 10, 2015. [Online]. Available: http://arxiv.org/abs/1501.02393

Supplementary material

A. Estimating second-order moments using Deep Sets and contrastive learning

A series of experiments was conducted to test if the Deep Sets model, as proposed in the main study, can identify different populations based on their second-order moments. In the first experiment, a toy dataset of n different populations was used. The mean of each population was factored out and the model was trained to distinguish the different populations. The second experiment repeated this setup, but used the real single-cell feature data instead, as described in the Experimental Setup. A final experiment was then conducted with the toy dataset to test the degree to which the model can learn to infer second-order moments from the input data. This was achieved by first factoring out the mean of the populations and then gradually sphering them.

The toy dataset was created using the following steps. For each of the n classes, a different low dimensional covariance matrix ($d \times d$) was created. Then, for each class, m random d dimensional points were generated from a standard normal distribution and rotated by calculating the dot product with their respective covariance matrix. By sampling all of the points from a standard normal distribution, all of their populations inherently have the same mean. The populations are then standardized, to make sure the model is learning the covariance and not the variance. Finally, k samples of q points are taken from each class, to simulate replicate populations as in the real data used in this study.

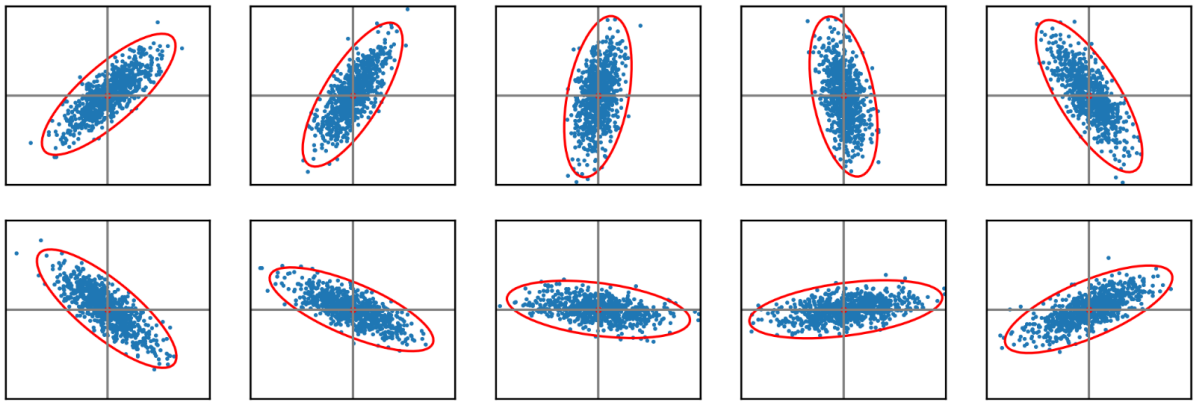


Figure A1: Two-dimensional point set distributions for ten different population classes. The covariance distributions are indicated using red ellipsoids.

For the first experiment, n, d, m, q, and k are chosen to be 10, 2, 3200, 800, and 4 respectively, *Figure A1*. The training setup was similar to the setup used for the real single-cell feature data, which is described in the Experimental Setup. Six of the 10 classes were used to train the model and four were used as the test set. There was no need for a validation set, as no optimization for the model was done. The model architecture was generally the same as the one used for the real data, although the number of nodes per layer was drastically reduced. The input layer, first projection layer, and last projection layer consisted of 64, 32, and 2 nodes respectively. The optimizer and loss function were identical. The model was trained for 20 epochs with a batch size of 6.

The model was compared to the average profile, which should give random results, by calculating the mAP for finding replicate populations in the test set. If the model is able to complete this task better than this baseline, that proves that this architecture is able to learn to infer covariance matrices from the input data. Although not irrefutable, it makes it more likely that the model for the single-cell feature data in this study is doing this as well.

After training the model in the described way, a mAP of 0.95 was achieved for finding replicate populations in the test set. This was higher than the baseline mAP of 0.31, which is similar to randomly choosing a sample in this test set. This means that the model is able to learn to infer the covariance matrices or some other higher-order statistic from the input data.

The second experiment aimed to verify these results by repeating the experiment on single-cell feature data from 11 Stain3 plates. Three plates were used as training plates and the eight other plates were used as a test set. The compounds were split in an 80% training and 20% validation set in the same way across all plates, just as described in the Experimental Setup. The cells are feature-averaged per well, which means that the average profile should give random results again. The model is trained in the same way as in the main study, except for the number of epochs which was set to 40 instead of 100. The results are shown in *Table A1*.

The trained model achieved an average mAP of 0.30 and 0.23 across all plates on the training and validation compounds of the test set respectively. These were both higher than their respective baseline mAP of 0.03 and 0.04, which was random. This proves that the model is also able to learn to infer the covariance matrices or some other higher-order statistic from real high-dimensional input data.

The third and final experiment tested the degree to which the model can learn to infer second-order moments from the input data. In this experiment, the same toy dataset was used as in the first experiment, but now the populations are also gradually sphered. Sphering can be applied to a certain degree, using the regularization hyperparameter r. Here, a distinction is made between low ($r = 0.3$), medium ($r = 0.1$), and high ($r = 0.01$) sphering. Low sphering leaves much of the second-order moments intact, while high sphering removes nearly all of that information from the population. The mAP was calculated in the same way as in the first experiment. The model was trained in the same way as well, however, a smaller learning rate (10^{-6} instead of 5×10^{-4}) and more epochs (1000 instead of 20) were required to train the

model on this data. The different degrees of sphering are shown for one of the 10 classes in *Figure A2*.

Table A1. Results of the second experiment. The mAP of the model and the baseline are shown for each training and test plate, and for the training and test compounds separately.

plate	**training compounds mAP model**	**training compounds mAP baseline**	**test compounds mAP model**	**test compounds mAP baseline**
Training plates				
BR00115134	0.44	0.03	0.24	0.04
BR00115125	0.37	0.03	0.25	0.04
BR00115133highexp	0.38	0.02	0.17	0.02
Test plates				
BR00115128highexp	0.33	0.03	0.25	0.04
BR00115125highexp	0.29	0.03	0.22	0.02
BR00115131	0.32	0.03	0.22	0.03
BR00115133	0.32	0.03	0.16	0.04
BR00115127	0.29	0.03	0.22	0.05
BR00115128	0.33	0.03	0.29	0.04
BR00115129	0.3	0.03	0.26	0.04
BR00115126	0.2	0.03	0.22	0.04

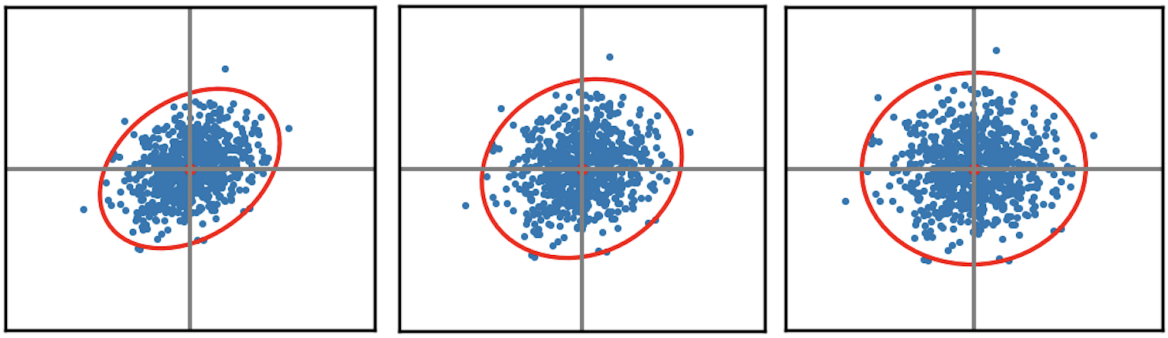


Figure A2. Left: low sphering, middle: medium sphering, and right: heavy sphering on one of the 10 classes of the toy dataset.

Table A2. Results of the third experiment. The mAP of the model and baseline are shown for each degree of sphering, based on the sphering regularization parameter r.

Degree of sphering	**mAP model**	**mAP baseline**
low ($r = 0.3$)	0.72	0.25
medium ($r = 0.1$)	0.61	0.25
high ($r = 0.01$)	0.29	0.25

The mAP of the model is about random after heavy sphering of the data, indicating that the model is no longer able to learn how to discern the different classes, *Table A2*. Even after varying a few hyperparameters like the model width, learning rate, and number of epochs the mAP did not improve. This is expected because the data can be almost fully explained by its second-order moments, which sphering factors out. It is not likely that this is the case for real single-cell feature data, where more interactions are expected.
After medium or low sphering, the model is still able to beat the baseline, although it requires many more training steps and a smaller learning rate than in the first experiment. In fact, the degree to which second-order moments are present correlates with the mAP achieved by the model.

These experiments show that the model is indeed learning to infer second-order moments from the input data. The different experiments also quantify the complexity of the task as the number of epochs or learning rate had to be adjusted to properly train the model. It was easier for the model to learn second-order moments from zero-meaned and standardized two dimensional feature data than high dimensional ($d = 1324$) feature data. Additionally, as the presence of the second-order moments was gradually decreased in the input data the model's ability to learn gradually decreased as well.

B. Training, validation, and test set stratification

Table B1: The training, validation, and test set stratification for Stain2, Stain3, Stain4, and Stain5. Five training, four validation, and three test plates are used for Stain2, Stain3, and Stain4. Stain5 contains six test set plates only.

Stain2	**Stain3**	**Stain4**	**Stain5**
Training plates			*Test plates*
BR00113818	BR00115128	BR00116627	BR00120532
BR00113820	BR00115125highexp	BR00116631	BR00120270
BR00112202	BR00115133highexp	BR00116625	BR00120536
BR00112197binned	BR00115131	BR00116630highexp	BR00120530

BR00112198	BR00115134	200922_015124-Vhighexp	BR00120526
Validation plates			BR00120274
BR00112197standard	BR00115129	BR00116628highexp	
BR00112197repeat	BR00115133	BR00116629highexp	
BR00112204	BR00115128highexp	BR00116627highexp	
BR00112201	BR00115127	BR00116629	
Test plates			
BR00112199	BR00115134bin1	200922_044247-Vbin1	
BR00113819	BR00115134multiplane	200922_015124-V	
BR00113821	BR00115126highexp	BR00116633bin1	

C. Model hyperparameter overview

Table C1: Overview of all tunable hyperparameters and their chosen values based on a random search hyperparameter optimization.

Hyperparameter	**Chosen value**
Adam W learning rate	5×10^{-4}
AdamW weight decay	10^{-2}
epochs	100
number of different compounds per batch	4
number of samples per compound	18
batch size	$4 \times 18 = 72$
Gaussian mean (for sampling number of cells)	1500
Gaussian standard deviation (for sampling number of cells)	800
latent dimension after first layer	2048
latent dimension after first projection layer	256
number of projection layers	2
output dimension of model (loss/aggregated profile space)	2048
SupCon loss temperature	0.1

D. Relevance correlation with CellProfiler features

Table D1: Top 20 CellProfiler features based on their negative Pearson correlation coefficient with the SA and CPA combined relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	Feature name	Correlation
Cells	Cells_Intensity_MeanIntensityEdge_DNA	-0.74
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_DNA	-0.72
Cytoplasm	Cytoplasm_Intensity_UpperQuartileIntensity_DNA	-0.71
Cytoplasm	Cytoplasm_Intensity_MeanIntensity_DNA	-0.69
Cytoplasm	Cytoplasm_Correlation_K_Brightfield_DNA	-0.67
Cells	Cells_Intensity_MedianIntensity_DNA	-0.64
Cells	Cells_Intensity_MeanIntensity_DNA	-0.63
Cells	Cells_Intensity_MeanIntensityEdge_ER	-0.61
Cytoplasm	Cytoplasm_Correlation_K_Mito_DNA	-0.61
Cytoplasm	Cytoplasm_Intensity_StdIntensity_DNA	-0.61
Cytoplasm	Cytoplasm_Intensity_MedianIntensity_DNA	-0.61
Cytoplasm	Cytoplasm_Intensity_MADIntensity_DNA	-0.61
Cells	Cells_Intensity_MeanIntensityEdge_RNA	-0.6
Cells	Cells_Intensity_MeanIntensityEdge_AGP	-0.6
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_RNA	-0.6
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_ER	-0.59
Cytoplasm	Cytoplasm_RadialDistribution_RadialCV_DNA_3of4	-0.58
Cells	Cells_Intensity_LowerQuartileIntensity_DNA	-0.58
Cells	Cells_Intensity_MaxIntensityEdge_RNA	-0.58
Cells	Cells_Intensity_MaxIntensityEdge_ER	-0.57

Table D2: Top 20 CellProfiler features based on their negative Pearson correlation coefficient with the SA and CPA combined relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	Feature name	Correlation
Cytoplasm	Cytoplasm_Correlation_K_DNA_Brightfield	0.72
Cells	Cells_AreaShape_MeanRadius	0.71
Cells	Cells_AreaShape_MaximumRadius	0.7
Cells	Cells_AreaShape_MedianRadius	0.7
Cells	Cells_AreaShape_Area	0.68
Cells	Cells_AreaShape_MinorAxisLength	0.68
Cells	Cells_AreaShape_MinFeretDiameter	0.68
Cytoplasm	Cytoplasm_AreaShape_MinFeretDiameter	0.68
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensityEdge_Brightfield	0.68
Cells	Cells_Intensity_IntegratedIntensity_Brightfield	0.67
Cytoplasm	Cytoplasm_AreaShape_Perimeter	0.67
Cytoplasm	Cytoplasm_AreaShape_MinorAxisLength	0.66
Cytoplasm	Cytoplasm_AreaShape_Area	0.65
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensity_Brightfield	0.65
Cells	Cells_AreaShape_Perimeter	0.64
Nuclei	Nuclei_AreaShape_MedianRadius	0.64
Cells	Cells_Intensity_IntegratedIntensityEdge_Brightfield	0.64

Nuclei	Nuclei_AreaShape_MeanRadius	0.64
Cytoplasm	Cytoplasm_AreaShape_MedianRadius	0.64
Cytoplasm	Cytoplasm_AreaShape_MeanRadius	0.64

Table D3: Top 20 CellProfiler features based on their negative Pearson correlation coefficient with the SA relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	**Feature name**	**Correlation**
Cells	Cells_Intensity_MeanIntensityEdge_DNA	-0.69
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_DNA	-0.67
Cytoplasm	Cytoplasm_Intensity_UpperQuartileIntensity_DNA	-0.67
Cytoplasm	Cytoplasm_Correlation_K_Brightfield_DNA	-0.66
Cytoplasm	Cytoplasm_Intensity_MeanIntensity_DNA	-0.66
Cells	Cells_Intensity_MeanIntensity_DNA	-0.61
Cells	Cells_Intensity_MedianIntensity_DNA	-0.60
Cells	Cells_Intensity_MaxIntensityEdge_RNA	-0.58
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_RNA	-0.58
Cytoplasm	Cytoplasm_Intensity_MADIntensity_DNA	-0.58
Cytoplasm	Cytoplasm_Intensity_MedianIntensity_DNA	-0.58
Cells	Cells_Intensity_MaxIntensityEdge_ER	-0.58
Cells	Cells_Intensity_MeanIntensityEdge_ER	-0.58
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_ER	-0.58
Cytoplasm	Cytoplasm_Intensity_StdIntensity_DNA	-0.57
Cells	Cells_Intensity_MeanIntensityEdge_AGP	-0.57
Cells	Cells_Intensity_MeanIntensityEdge_RNA	-0.57
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_AGP	-0.56
Cytoplasm	Cytoplasm_RadialDistribution_RadialCV_DNA_3of4	-0.56
Nuclei	Nuclei_RadialDistribution_RadialCV_DNA_4of4	-0.55

Table D4: Top 20 CellProfiler features based on their positive Pearson correlation coefficient with the SA relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	**Feature name**	**Correlation**
Cytoplasm	Cytoplasm_Correlation_K_DNA_Brightfield	0.68
Cells	Cells_AreaShape_MeanRadius	0.63
Cells	Cells_AreaShape_MaximumRadius	0.62
Cells	Cells_AreaShape_MedianRadius	0.62
Cells	Cells_AreaShape_MinorAxisLength	0.60
Cytoplasm	Cytoplasm_AreaShape_MinFeretDiameter	0.60
Cells	Cells_AreaShape_MinFeretDiameter	0.60
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensityEdge_Brightfield	0.59
Nuclei	Nuclei_AreaShape_MedianRadius	0.58
Cells	Cells_AreaShape_Area	0.58
Nuclei	Nuclei_AreaShape_MeanRadius	0.58
Cells	Cells_Intensity_IntegratedIntensity_Brightfield	0.58
Cytoplasm	Cytoplasm_AreaShape_Perimeter	0.58
Cytoplasm	Cytoplasm_AreaShape_MinorAxisLength	0.57
Cytoplasm	Cytoplasm_AreaShape_MeanRadius	0.57
Cells	Cells_Neighbors_FirstClosestDistance_Adjacent	0.57
Cytoplasm	Cytoplasm_AreaShape_MedianRadius	0.57

Cytoplasm	Cytoplasm_AreaShape_MaximumRadius	0.56
Cells	Cells_Intensity_IntegratedIntensityEdge_Brightfield	0.56
Cytoplasm	Cytoplasm_AreaShape_Area	0.56

Table D5: Top 20 CellProfiler features based on their negative Pearson correlation coefficient with the CPA relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	**Feature name**	**Correlation**
Cytoplasm	Cytoplasm_Correlation_K_Mito_DNA	-0.51
Cells	Cells_Intensity_MeanIntensityEdge_DNA	-0.49
Cells	Cells_Correlation_Overlap_Mito_RNA	-0.48
Cytoplasm	Cytoplasm_Intensity_MeanIntensityEdge_DNA	-0.48
Cells	Cells_RadialDistribution_MeanFrac_Mito_4of4	-0.47
Cytoplasm	Cytoplasm_Correlation_Overlap_Mito_RNA	-0.47
Cytoplasm	Cytoplasm_Correlation_K_Mito_AGP	-0.46
Cytoplasm	Cytoplasm_Correlation_Manders_Brightfield_Mito	-0.46
Cells	Cells_Correlation_Manders_Brightfield_Mito	-0.46
Cells	Cells_Correlation_Overlap_AGP_Brightfield	-0.46
Cytoplasm	Cytoplasm_Correlation_Manders_RNA_Mito	-0.45
Cytoplasm	Cytoplasm_Correlation_Manders_ER_Mito	-0.45
Cytoplasm	Cytoplasm_Correlation_Manders_AGP_Mito	-0.45
Cells	Cells_Correlation_Manders_ER_Mito	-0.45
Cells	Cells_Correlation_Manders_RNA_Mito	-0.45
Cytoplasm	Cytoplasm_Correlation_K_Mito_RNA	-0.45
Cells	Cells_Correlation_Manders_AGP_Mito	-0.45
Cytoplasm	Cytoplasm_Correlation_Manders_DNA_Mito	-0.45
Cells	Cells_Correlation_K_Mito_AGP	-0.45
Cytoplasm	Cytoplasm_Intensity_UpperQuartileIntensity_DNA	-0.45

Table D6: Top 20 CellProfiler features based on their positive Pearson correlation coefficient with the CPA relevance score. The scores were calculated for a single test plate of Stain3 (200922_015124-V).

Feature category	**Feature name**	**Correlation**
Cells	Cells_Intensity_IntegratedIntensity_Mito	0.67
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensity_Mito	0.65
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensityEdge_Mito	0.60
Cells	Cells_AreaShape_Area	0.59
Cells	Cells_Intensity_IntegratedIntensity_Brightfield	0.58
Cytoplasm	Cytoplasm_AreaShape_Area	0.57
Cells	Cells_Intensity_IntegratedIntensity_AGP	0.57
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensity_Brightfield	0.56
Nuclei	Nuclei_Intensity_IntegratedIntensityEdge_Mito	0.56
Cytoplasm	Cytoplasm_AreaShape_Perimeter	0.56
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensity_DNA	0.56
Cells	Cells_AreaShape_MaximumRadius	0.55
Cells	Cells_Intensity_StdIntensity_Mito	0.55
Cells	Cells_AreaShape_MeanRadius	0.55
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensity_AGP	0.55
Cytoplasm	Cytoplasm_Intensity_IntegratedIntensityEdge_Brightfield	0.55
Cytoplasm	Cytoplasm_Correlation_K_DNA_Mito	0.55

Cells	Cells_AreaShape_MedianRadius	0.55
Cells	Cells_AreaShape_Perimeter	0.55
Cells	Cells_Intensity_IntegratedIntensity_DNA	0.54

E. Measuring plate similarity

Plate similarity is measured using a hierarchical clustering analysis. First, the average profile is taken per well for each available plate. This results in 384 well profiles of 1324 features per plate. PCA analysis is then performed on this matrix and the loadings of the first principal component (PC1) are taken, resulting in a 1x1324 vector. This vector is then normalized to a unit vector. These vectors are calculated for all available plates and the Pearson correlation is calculated between them. Comparing the PC1 loadings of two multivariate distributions is an approximation for comparing their covariance matrices [34], and thus the Pearson correlation can quantify their similarity. Finally, the Pearson correlation is used to create a hierarchical clustering map, Figure E1. The clustermap shows that Stain5 and Stain2, Stain3, and Stain4 can be separated as the two main clusters, signifying the strong experiment effects between these two groups. Plates that do not lie within a cluster of 6 or more plates in this cluster map are considered out-of-distribution plates, based on their batch effects.

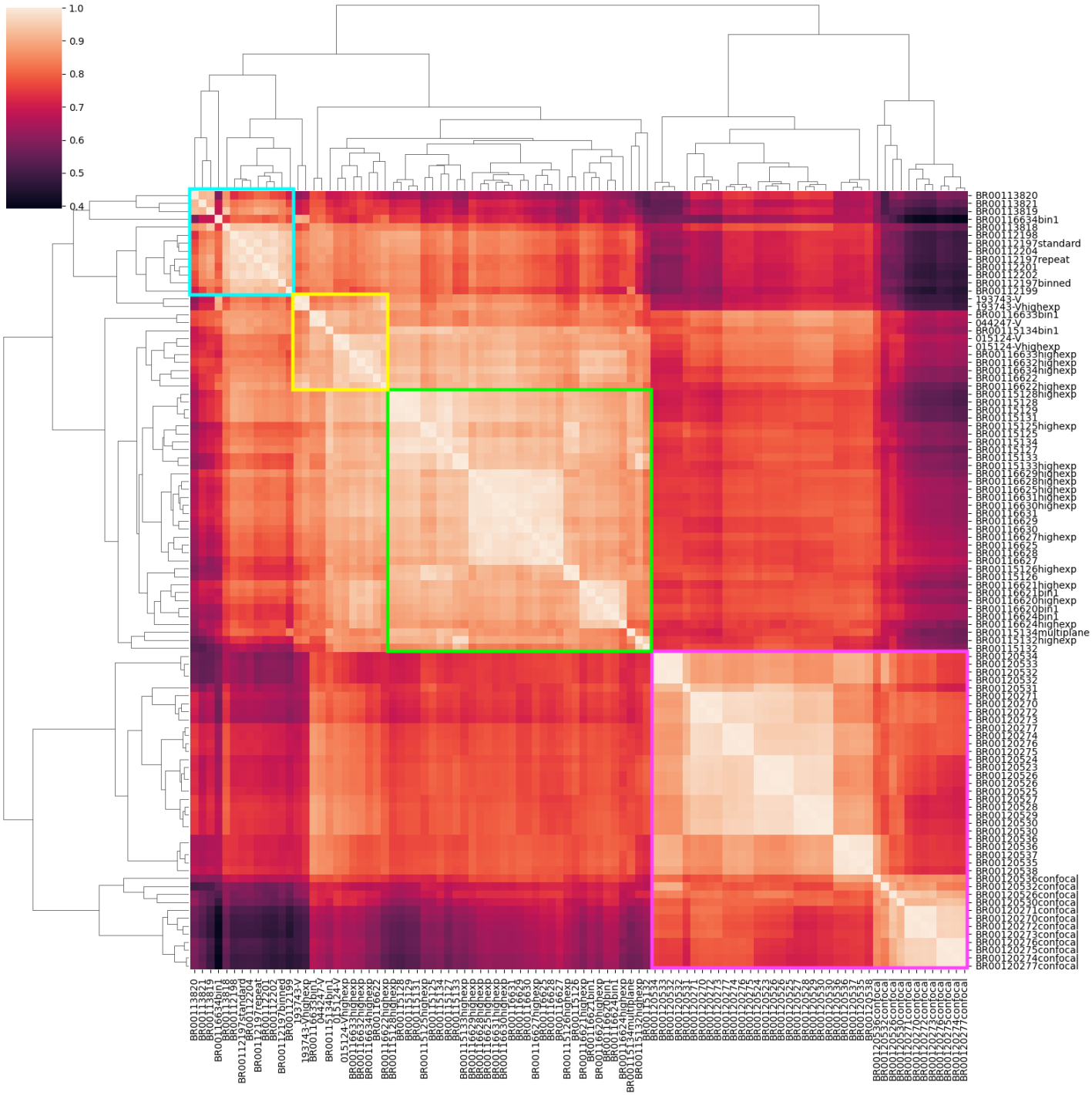
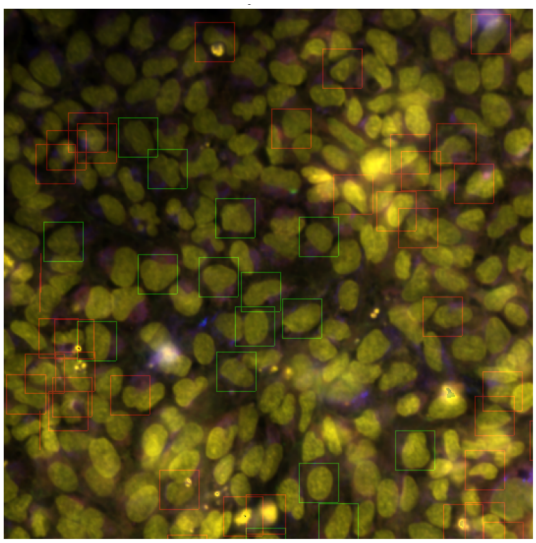


Figure E1: Hierarchical clustermap of the Pearson correlations between the PC1 loadings of the mean aggregated profile features per plate. The plate clusters of Stain2 (cyan), Stain3 (yellow), Stain4 (green), and Stain5 (pink) are annotated with boxes.

F. Additional cell image FOVs



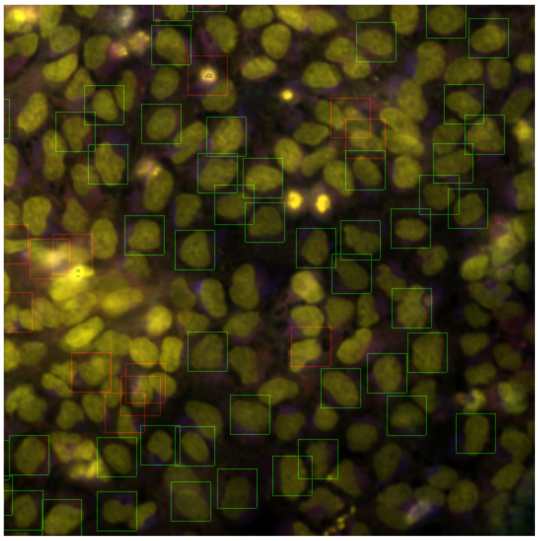


Figure F1: 5-channel combined microscope image of the second and third FOVs for plate BR00112197binned. The most relevant cells are annotated with green boxes and the least relevant cells are annotated with red boxes.

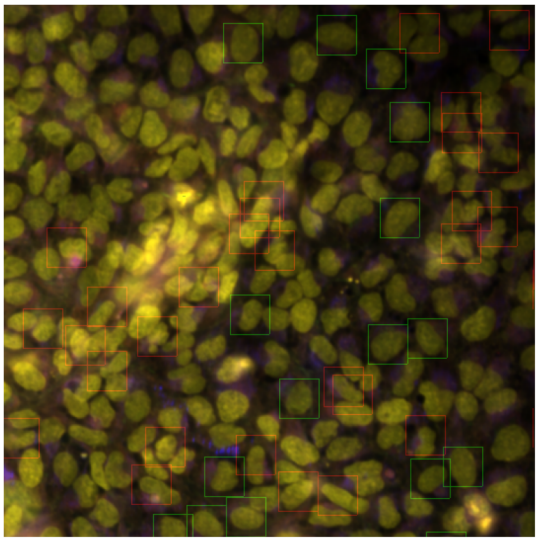


Figure F2: 5-channel combined microscope image of the fourth FOV for plate BR00112197binned. The most relevant cells are annotated with green boxes and the least relevant cells are annotated with red boxes.