Learning representations for image-based profiling of perturbations

Authors

Nikita Moshkov, Michael Bornholdt, Santiago Benoit, Matthew Smith, Claire McQuin, Allen Goodman, Rebecca Senft, Yu Han, Mehrtash Babadi, Peter Horvath, Beth A. Cimini, Anne E. Carpenter, Shantanu Singh, Juan C. Caicedo*

* Corresponding author

Abstract

Measuring the phenotypic effect of treatments on cells through imaging assays is an efficient and powerful way of studying cell biology, and requires computational methods for transforming images into quantitative data that highlights phenotypic outcomes. Here, we present an optimized strategy for learning representations of treatment effects from high-throughput imaging data, which follows a causal framework for interpreting results and guiding performance improvements. We use weakly supervised learning (WSL) for modeling associations between images and treatments, and show that it encodes both confounding factors and phenotypic features in the learned representation. To facilitate their separation, we constructed a large training dataset with Cell Painting images from five different sources to maximize experimental diversity, following insights from our causal analysis. Training a WSL model with this dataset successfully improves downstream performance, and produces a reusable convolutional network for image-based profiling, which we call *Cell Painting CNN*. We conducted a comprehensive evaluation of our strategy on three publicly available Cell Painting datasets, discovering that representations obtained by the Cell Painting CNN can improve performance in downstream analysis up to 25% with respect to classical features, while also being more computationally efficient.

Introduction

High-throughput imaging and automated image analysis are powerful tools for studying the inner workings of cells under experimental interventions. In particular, the Cell Painting assay [1,2] has been adopted both by academic and industrial laboratories to evaluate how perturbations alter overall cell biology. It has been successfully used for studying compound libraries [3–5], predicting phenotypic activity [6–9], and profiling human disease [10,11], among many others. To reveal the phenotypic outcome of treatments, image-based profiling transforms microscopy images into rich high-dimensional data using morphological feature extraction [12]. Cell Painting datasets with thousands of experimental interventions provide a unique opportunity to use machine learning for obtaining causal representations of the phenotypic outcomes of treatments.

Improved feature representations of cellular morphology have the potential to increase the sensitivity and robustness of image-based profiling to support a wide range of discovery applications [13,14]. Feature extraction has been traditionally approached with classical image processing [15,16], which is based on manually engineered features that may not capture all the relevant phenotypic variation. Several studies have used convolutional neural networks (CNNs) pre-trained on natural images [17–19], which are optimized to capture variation of macroscopic objects instead of images of cells. To recover causal representations of treatment effects, feature representations need to be sensitive to subtle changes in morphology. However, classical features and pre-trained networks may not have sufficient expressive power to realize that potential. Representation learning has been used as a tool to learn domain-specific features from cellular images in a data-driven way [4,20–23], but it also brings unique challenges to avoid being dominated by confounding factors [24,25].

In this paper, we investigate the problem of learning representations for image-based profiling with Cell Painting. Our goal is to identify an optimal strategy for learning cellular features, and then use it for training models that recover improved representations of the phenotypic outcomes of treatments. We use a causal framework to reason about the challenges of learning representations of cell morphology (e.g. confounding factors), which naturally fits in the context of perturbation experiments [26,27], and serves as a tool to optimize the workflow and yield better performance (Figure 1). In addition, we adopted a quantitative evaluation of the impact of feature representations in a biological downstream task, to guide the search for an optimized workflow. The evaluation is based on querying a reference collection of treatments to find biological matches in a perturbation experiment. In each evaluation, cell morphology features change to compare different strategies, while the rest of the image-based profiling workflow remains constant. Performance is measured using metrics for the quality of a ranked list of results for each query (Figure 1H). With this evaluation framework, we conduct an extensive analysis on three publicly available Cell Painting datasets.

Within the proposed causal framework, weakly supervised learning (WSL) [20] is a powerful approach to model the associations between images and treatments, and we found that it captures rich cellular features that simultaneously encode confounding factors and phenotypic

outcomes as latent variables. Our analysis indicates that to disentangle them and recover the causal representations of the true outcome of perturbations, it is important to train models with highly diverse data for improving the ability of models to learn the difference between the two types of variation. Therefore, we constructed a new dataset that combines variation from five different sources to maximize the diversity of treatments and confounding factors (Figure 1F). As a result, we successfully trained a new, reusable model: the Cell Painting CNN (Figure 1G), which yields better performance in the evaluated benchmarks and displays sufficient generalization ability to profile other datasets effectively.

Results

Recovering features of the causal effects of treatments

We use a causal model as a conceptual framework to reason and analyze the results of representation learning strategies. Figure 1B presents the causal graph with four variables: interventions (treatments $\mathbf{T}$), observations (images $\mathbf{O}$), outcomes (phenotypes $\mathbf{Y}$) and confounders (e.g. batches $\mathbf{C}$). For simplicity, we assume that there is a single context ($\mathbf{X}$, not in the diagram) for experimental treatments, which are clonal cells of an isogenic cell line; perturbation experiments with multiple cell lines may need a different model. Some variables are observables (shaded circles), while others represent latent variables (empty circles). This graph is a model of the causal assumptions we make for representation learning and for interpreting the results.

We are interested in estimating a continuous, multidimensional representation of treatment outcomes ($\mathbf{Y}$), which can later be used in downstream analysis tasks. A major issue for learning a causal representation of the phenotypic outcome of treatments is technical variation ($\mathbf{C}$), which groups a set of factors introducing unwanted variation in images. Note that images, phenotypes and treatments respond to changes in technical variation. Images are impacted by artifacts in image acquisition, including microscope settings and assay preparation. Treatments are impacted by plate map designs that are not fully randomized and usually group treatments in specific plate positions. Confounders also impact phenotypes due to variations in cell density and other conditions that make cells grow and respond differently.

The main direct cause of phenotypic changes in cells in a perturbation experiment is the treatment. We observe treatment outcomes indirectly through imaging assays, and thus, we need image analysis to recover the phenotypic effect and to separate it from unwanted variation. A representation of the phenotypic effect can be obtained with the workflow depicted in Figures 1C-E, which illustrates three major steps: 1) modeling the correlations between images and treatments using a CNN trained with weakly supervised learning (WSL), 2) using batch correction to learn a transformation of the latent representation of images obtained from intermediate layers of the CNN, and 3) generating representations of treatment effects in cellular morphology for downstream analysis.

We use WSL [20] (Figure 1C) for obtaining representations of the phenotypic outcome of treatments by training a classifier to distinguish all treatments from each other. This approach models the relationship between observed images and treatments using an EfficientNet [28] with a classification loss (Methods), and makes the following assumptions: 1) if a treatment has an observable effect then it can be seen in images, and therefore, training a CNN helps identify visual features that make it detectably different from all other treatments. 2) Treatment labels in the classification task are weak labels because they are not the final outcome of interest, they do not reflect expert biological ground truth, and there is no certainty that all treatments produce a phenotypic outcome, nor that they produce a different phenotypic outcome from each other. 3) Intermediate layers of the CNN trained with treatment labels capture all visual variation of images as latent variables, including confounders and causal phenotypic features.

To recover the phenotypic features of treatments from the latent representations of the weakly supervised CNN, we employ a batch correction model inspired by the Typical Variation Normalization (TVN) technique [18]. This transform aims to reduce the variation associated with confounders and amplify features caused by phenotypic outcomes (Figure 1D). The main idea of this approach is to use negative control samples as a model of unwanted variation under the assumption that their phenotypic features should be neutral, and therefore differences in control images reflect mainly confounding factors. We follow this assumption and use a sphering transformation (Methods) to learn a function that projects latent features from the CNN to a corrected feature space that preserves the phenotypic features caused by treatments.

As a result of learning latent features with WSL and correcting for unwanted variation, the new feature space more cleanly highlights feature representations of treatment effects. These representations are continuous multidimensional estimators of the true phenotypic outcome of treatments in the perturbation experiment, and because of their unbiased design can be used to approach various questions and downstream analysis tasks, such as predicting compound bioactivity or inferring the impact of cancer variants (Figure 1E). In this paper, we conduct a quantitative evaluation of these representations in a biological matching task (genetic pathway or MoA) using a guilt by association approach, which is a meaningful way to validate the biological relevance of the resulting features.

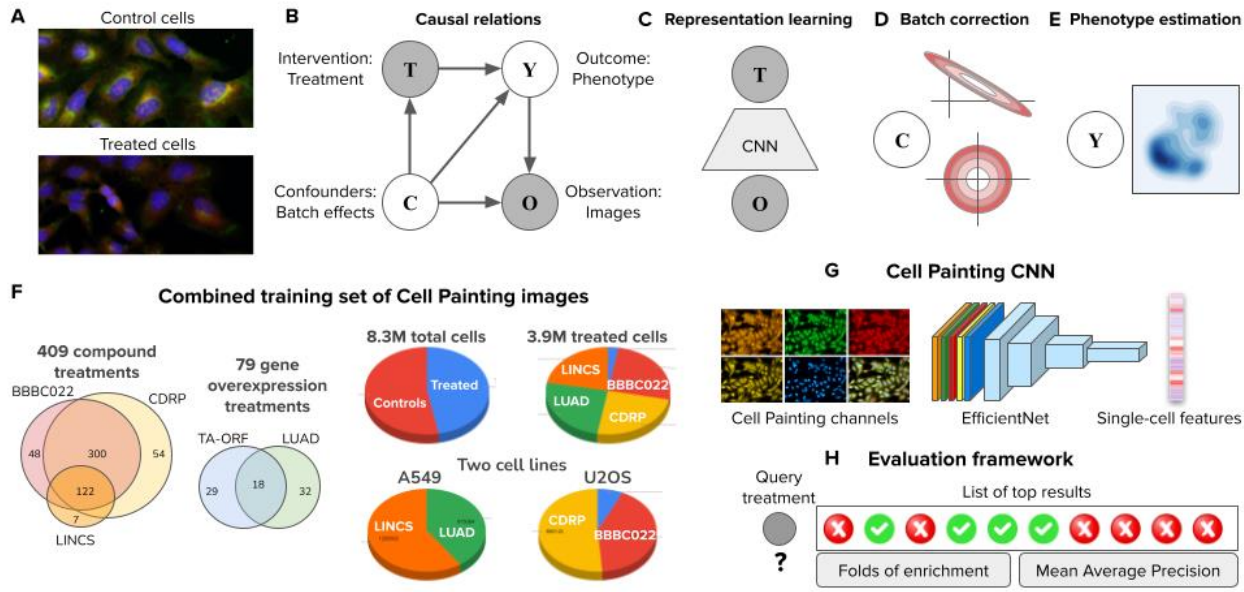


Figure 1. Framework for analyzing image-based profiling experiments. *A) Example Cell Painting images from the BBBC037 dataset of control cells (empty status) and one experimental intervention (JUN wild-type overexpression) in the U2OS cell line. B) Causal graph of a conventional high-throughput Cell Painting experiment with two observables in shaded circles (treatments and images) and two latent variables in empty circles (phenotypes and batch effects). The arrows indicate the direction of causation. C) Weakly supervised learning as a strategy to model associations between images (*$\boldsymbol{O}$*) and treatments (*$\boldsymbol{T}$*) using a convolutional neural network (CNN). The CNN captures information about the latent variables* $\boldsymbol{C}$ *and* $\boldsymbol{Y}$ *in the causal graph because both are intermediate nodes in the paths connecting images and treatments. D) Illustration of the sphering batch-correction method where control samples are a model of unwanted variation (top). After sphering, the biases of unwanted variation in control samples is reduced (bottom). E) The goal of image-based profiling is to recover the outcome of treatments by estimating a representation of the resulting phenotype, free from unwanted confounding effects. F) Statistics of the Combined Cell Painting dataset created to train a generalist model, which brings 488 treatments from 5 different publicly available sources (Methods): LINCS, LUAD, and the three datasets evaluated here; the Venn diagrams illustrate the common treatments among them. There are two types of treatments (compounds and gene overexpression), two types of controls (empty and DMSO), two cell lines (A549 and U2OS), for a total of 8.3 million single cells from 232 plates in the resulting training resource. G) Illustration of the Cell Painting CNN, an EfficientNet model trained on the dataset in F to extract features from single cells. H) The evaluation of performance is based on nearest neighbor queries performed in the space of phenotype representations to match treatments with the same phenotypic outcome. Performance is measured with two metrics: folds of enrichment and mean average precision (Methods).*

Representations learned by the Cell Painting CNN encode improved biological features

Figure 2C shows a UMAP projection [29] of the feature space obtained using our Cell Painting CNN for the three datasets evaluated in this study. From a qualitative perspective, the UMAP plot of the BBBC037 dataset (a gene overexpression screen) shows groups of treatments clustered according to their corresponding genetic pathway, and recapitulates previous observations of known biology [30]. The BBBC022 [31] and BBBC036 [32] datasets (compound

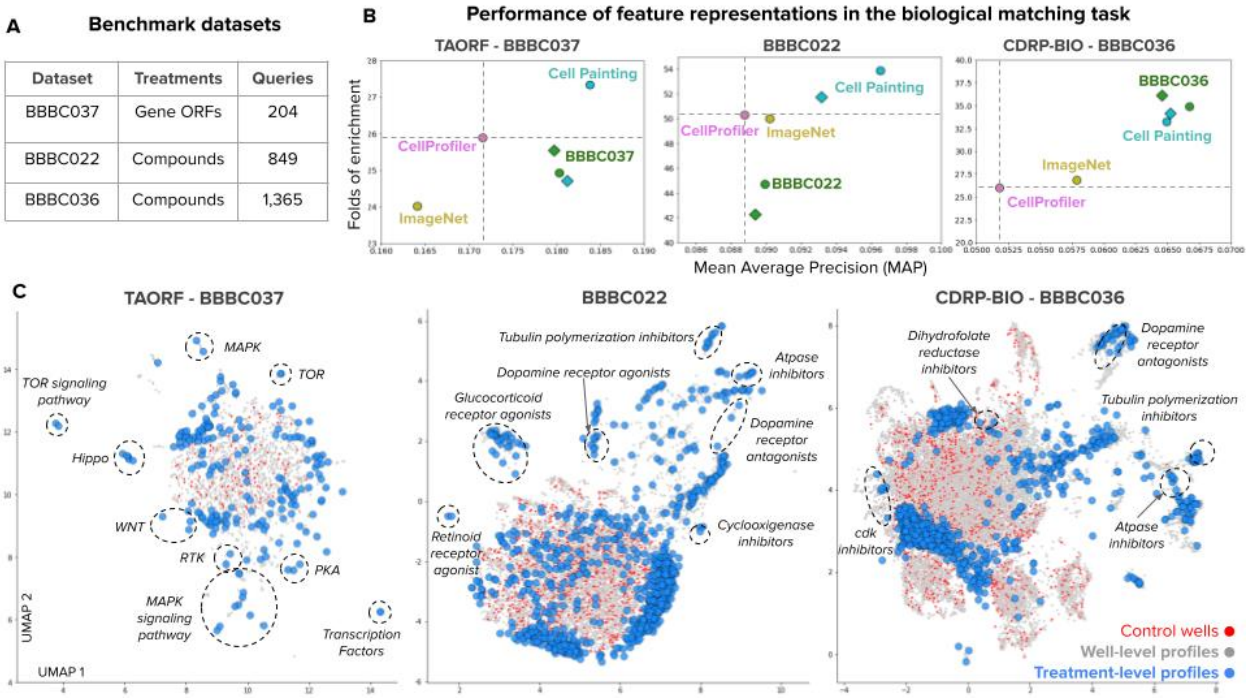
screens) likewise show many treatments grouped together according to their mechanism of action (MoA).

For quantitative comparison of multiple feature extraction strategies, we simulate a user searching a reference library to find a “match” to their query treatment of interest. We used a leave-one-treatment-out strategy for all annotated treatments in the three benchmark datasets, following previous research in the field [18,33,34]. In all cases, queries and reference items are aggregated treatment-level profiles (Methods) matched using the cosine similarity (Methods). The result of searching the library with one treatment query is a ranked list of treatments in descending order of relevance. A result in the ranked list is considered a positive hit if it shares at least one biological annotation in common with the query; otherwise it is a negative result (Figure 1H). Biological annotations are genetic pathways for overexpression treatments in BBBC037 and MoAs for compound treatments in the BBBC022 and BBBC036 datasets. Then, we proceed to quantify performance using the folds of enrichment and mean average precision metrics (see Methods).

The generalized Cell Painting CNN model, trained with WSL on the combined set, performs better than baseline approaches in the task of biologically matching queries against a reference annotated library of treatments, across all three benchmarks (Figure 2B). We consider two baseline strategies in our evaluation: 1) creating image-based profiles with classical features obtained with custom CellProfiler pipelines (Methods), and 2) computing profiles with a CNN pre-trained on ImageNet, a dataset of natural images in RGB (Methods). Intuitively, we expect feature representations trained on Cell Painting images to perform better at the matching task than the baselines. In the case of CellProfiler, manually engineered features may not capture all the relevant phenotypic variation, and in the case of ImageNet pre-trained networks, they are optimized for macroscopic objects in 3-channel natural images instead of 5-channel fluorescence images of cells.

We found that ImageNet features showed variable performance compared to CellProfiler features (Figure 2B), sometimes yielding similar performance (BBBC022), sometimes lower performance (BBBC037) and sometimes slightly better performance (BBBC036). The three benchmarks used in this study are larger scale and more challenging than datasets used in previous studies [17,18] where it was observed that ImageNet features are typically more powerful than classical features. Our results indicate that, in large scale perturbation experiments with Cell Painting, ImageNet features do not conclusively capture more cellular-specific variation than manually engineered features using classical image processing.

Finally, we also assessed models trained on each dataset alone, rather than our generalized Cell Painting CNN model trained on various datasets. We found that narrowing the training to only the dataset at hand typically diminished performance as compared to the generalized model (green dots in Figure 2B), with one exception, BBBC036, where they had comparable performance. This brings us to the question of what is necessary to train a successful Cell Painting model, which we explore in the next sections.



Dataset	Treatments	Queries
BBBC037	Gene ORFs	204
BBBC022	Compounds	849
BBBC036	Compounds	1,365

Figure 2. Quantitative evaluation of feature representations of treatment effects. *The main evaluation task is biological matching, which simulates queries with a treatment and aims to find other treatments that match their phenotypic outcome (e.g. same mechanism of action or same genetic pathway). A) Table of the three benchmark datasets used in this study. The quantitative results for each dataset are the mean across the queries listed in this table. B) Performance of feature representations for the three evaluated datasets according to two metrics: Mean Average Precision (MAP) in the x axis and Folds of Enrichment in the y axis (see Methods). Each point indicates the mean of these metrics over all queries using the following feature representations: CellProfiler (pink), a CNN pre-trained on ImageNet (yellow), a CNN trained on the combined set of Cell Painting images (cyan, Figure 1F), and a CNN trained on Cell Painting images from the same dataset (green). CNNs trained on Cell Painting images have two results depending on the training-validation strategy (see Figure 3 for more details): leave plates out (diamonds) or leave cells out (circles). C) UMAP visualizations of the phenotypic profiles of treatments recovered with the Cell Painting CNN after batch correction for the three datasets evaluated in this work. The plot includes a projection of well-level profiles (gray points), control wells (red points), and aggregated treatment-level profiles of treatments (blue points). Dashed lines indicate clusters of treatment-level profiles where all or the majority of the points share the same biological annotation.*

Weakly supervised learning captures latent representations of confounders and phenotypic outcomes of treatments

Since we observed that models trained on images from each dataset alone do not succeed at improving performance with respect to baselines (green points in Figure 2B) we sought to determine whether WSL models are capturing more information from confounding factors than phenotypic effects (variables $\mathbf{C}$ and $\mathbf{Y}$ in the causal graph, respectively). WSL models are trained with a classification loss to detect the treatment in images of single cells (Figure 1C, Methods), which is a pretext task to learn representations. Given that we always know the treatment applied to cells in a well, we can quantify the success rate of such classifiers on this

pretext task. We conducted single-cell classification experiments with WSL using two validation schemes, herein called leave-cells-out and leave-plates-out validation (Figure 3A), which aim to reveal how sensitive WSL is to biological and technical variation.

The leave-cells-out validation regime uses single cells from all plates and treatments in the experiment for training, and leaves a random fraction out for validation. By doing so, trained CNNs have the opportunity to observe the whole distribution of phenotypic features (all treatments) as well as the whole distribution of confounding factors (all batches or plates). In contrast, the leave-plates-out validation regime separates different technical replicates (plates) for training and validation, resulting in a model that still observes the whole distribution of treatments, but only partially sees the distribution of confounding factors.

The results in Figure 3B-D show a stark contrast in performance between the two validation strategies. When leaving cells out (results in blue), a CNN can accurately learn to classify single cells in the training and validation sets with only a minor difference in performance (Figure 3B); when leaving plates out (results in orange), the CNN learns to classify the training set well but fails to generalize correctly to the validation set, resulting in a major gap according to the learning curves (Figure 3B). The generalization ability of the two models is further highlighted in the validation results in Figures 3C,D, which present the distribution of precision, recall and F1-scores.

Importantly, while these WSL models trained on the same datasets alone exhibit a major difference in performance in the pretext classification task, their performance in the downstream analysis task is almost the same (green circles vs green diamonds in Figure 2B). In addition, WSL models trained on the full distribution of treatments usually fail to improve performance with respect to baseline approaches (green points in Figure 2B, except BBBC036). This dichotomy between performance in the pretext task and the downstream task reveals that WSL models leverage any way in which the relationship between images and treatments can be explained, including spurious correlations. In fact, the validation results of leaving-cells-out are an overly optimistic estimate of how well a CNN can recognize treatments in single cells, because the models leverage batch effects to make the correct connection. On the other hand, the results of leaving-plates-out are an overly pessimistic estimate because the CNN fails to generalize to unseen replicates with unknown confounding variation. The true estimate of performance in the pretext classification task is likely to be in the middle when accounting for confounding factors. This is indeed what we observe in the downstream analysis results: after batch correction, the representations of models trained with leave-cells-out and leave-plates-out come closer together in the biological matching task (green circles vs green diamonds in Figure 2B).

Figure 3. Validation strategies for the single-cell classification task in weakly supervised learning. *A) Illustration of the two strategies: leave-cells-out (in blue) uses cells from all plates in the dataset for training and leaves a random fraction out for validation. Leave-plates-out (in orange) uses cells from certain plates for training, and leaves entire plates out for validation. The difference in performance is primarily due to confounding factors. B) Learning curves of models trained with WSL for 30 epochs with all treatments from each dataset. The x-axis is the number of epochs and the y axis is the average F1-score. The color of lines indicates the validation strategy, and the style of lines indicates training (solid) or validation (dashed) data. C) Precision and recall results of each treatment in the single-cell classification task. Each point is a treatment (negative controls are labeled in blue), and the color corresponds to the validation strategy. D) Distribution of F1-scores for the single-cell classification task for each treatment: the x axis enumerates treatments (sorted by performance according to the leave-cells-out validation scheme), and the y-axis is the F1-score. The first and last treatments in the x axis are labeled.*

Treatments with strong phenotypic effect can improve performance

The WSL model depicted in Figure 1C captures direct associations between images ($\mathbf{O}$) and treatments ($\mathbf{T}$) in the causal graph, while encoding technical ($\mathbf{C}$) and phenotypic ($\mathbf{Y}$) variation as latent variables because both are valid paths to find correlations. Given that controlling the distribution of confounding factors does not change overall performance, in this section we explore the impact of controlling the distribution of phenotypic diversity. Our reasoning is that WSL learning favors correlations between treatments and images through the path in the causal graph that makes it easier to minimize the empirical error in the pretext task. Therefore, the variation of treatments with a weak phenotypic response is overpowered by confounding factors that are stronger relative to the phenotype.

We start by splitting treatments into two broad categories: treatments with strong phenotypic effect, and treatments with weak or no phenotypic effect. To measure the phenotypic strength of treatments we calculate the Euclidean distance between control and treatment profiles in the CellProfiler feature space after batch correction with a sphering transform. Then, we rank treatments in descending order and take a fraction of those most different from controls in the strong category (Methods). We interpret this procedure as a crude approximation of the average treatment effect (ATE), a causal parameter of intervention outcomes, because the Euclidean distance calculates the difference in expected values (aggregated profiles) of the outcome variable (phenotype) between the control and treated conditions. However, since we do not observe the control and treated condition in the same cells, this remains only an approximation of the ATE, even if the cells are isogenic clones of each other. We chose distances in the CellProfiler feature space as an independent prior for estimating treatment strength because these are non-trainable, and because in our experiments CellProfiler features exhibit more robustness to confounding factors (Supplementary Figure 2, and Supplementary Figure 3).

Next, we evaluated the performance of WSL models trained on strong treatments only and we found that performance tends to improve in the downstream biological matching task with respect to training with the full distribution of treatments in each dataset (Supplementary Figure 1). These results were obtained by training under a leave-plates-out validation regime, which also restricts the distribution of confounding factors. The trend indicates that it is possible to break the limitations of WSL for capturing useful associations between images and treatments in the latent variables without being overpowered by confounding factors. However, while the trend is generally positive, the results are still similar to baseline approaches, which suggests that training may need a higher diversity of strong treatments to reach improved performance.

Training on highly diverse experimental conditions yields improved representations

To increase the diversity of experimental conditions in the training set, we created a combined training resource by collecting strong treatments from five different dataset sources, including the three benchmarks evaluated in this work plus two additional publicly available Cell Painting datasets (Figure 1F, and Methods). In total, this combined set has 488 strong treatments, two types of negative controls, and examples from more than 200 plates, resulting in training data with high experimental diversity with respect to the two latent variables in the causal graph: high technical variation (confounders $\mathbf{C}$) and high phenotypic variation through strong treatments (outcomes $\mathbf{Y}$).

Training on this combined Cell Painting dataset of strong treatments consistently improves performance and yields better results than the baselines in all benchmarks (cyan points in Figure 2B). According to the MAP metric, a WSL model trained on the highly diverse combined set improves performance up to 7%, 8% and 25% relative to CellProfiler features on BBBC037, BBBC022 and BBBC036 respectively (difference of cyan points vs pink points in the x axis of Figure 2B). Similar improvements are observed in other metrics as well. Importantly, this

dataset allowed us to train a single model once and profile all the three benchmarks without re-training or fine-tuning on each of them, demonstrating that the model captures features of Cell Painting images relevant to distinguish more effectively the variation of the two latent variables of the causal model.

The dimensionality of the feature space of the Cell Painting CNN is also more compact than CellProfiler and the ImageNet CNN model (Supplementary Figure 4B), with the intermediate layer Conv6A of the EfficientNet B0 network being the best source of latent representations for downstream analysis in all of our experiments (Supplementary Figure 4D). This layer, after spatial average pooling, results in 672 features, compared to 1,700 of CellProfiler and 3,360 of CNN ImageNet (672 for each of the five imaging channels). The dimensionality of single-cell features has an impact on storage space, especially for large scale experiments, making the CNN Cell Painting an efficient choice too (Supplementary Figure 4C).

Learned representations separate factors of variation and facilitate batch correction

Batch-correction is a crucial step for image-based profiling regardless of the feature space of choice. We hypothesized that a rich representation might encode both confounders and phenotypic features in a way that facilitates separating one type of variation from the other, i.e. disentangles the sources of variation. To test this, we evaluated how representations respond to the sphering transform, a linear transformation for batch correction based on singular value decomposition SVD (Methods). Sphering first finds directions of maximal variance in the set of control samples and then reverses their importance by inverting the eigenvalues. This transform makes the assumption that large variation is associated with confounders and subtle variation is associated with phenotypes. Thus, sphering can succeed at recovering the phenotypic effects of treatments if the feature space separates the sources of variation in this way.

The results in Figure 4B show that all the methods benefit from batch correction with the sphering transform, indicated by the upward trend of all curves from low performance with no batch correction to improved performance as batch correction increases. Downstream performance in the biological matching task improves by about 50% on average when comparing against raw features without correction. The UMAP plots in Figure 4A show the Cell Painting CNN feature space for well-level profiles before batch correction. When colored by Plate IDs, the data points are fragmented, and the density functions in the two UMAP axes indicate concentration of plate clusters. After sphering, the UMAP plots in Figure 4C show more integrated data points and better aligned density distributions of plates. The performance of the Cell Painting CNN in the biological matching task also improves upon the baselines (Figure 4B) and displays a consistent ability to facilitate batch correction in all the three datasets, unlike the CNN - ImageNet.

The sphering transform, while effective, is still far from perfect, and further research is needed to better disentangle confounding from phenotypic variation, potentially using nonlinear transformations.

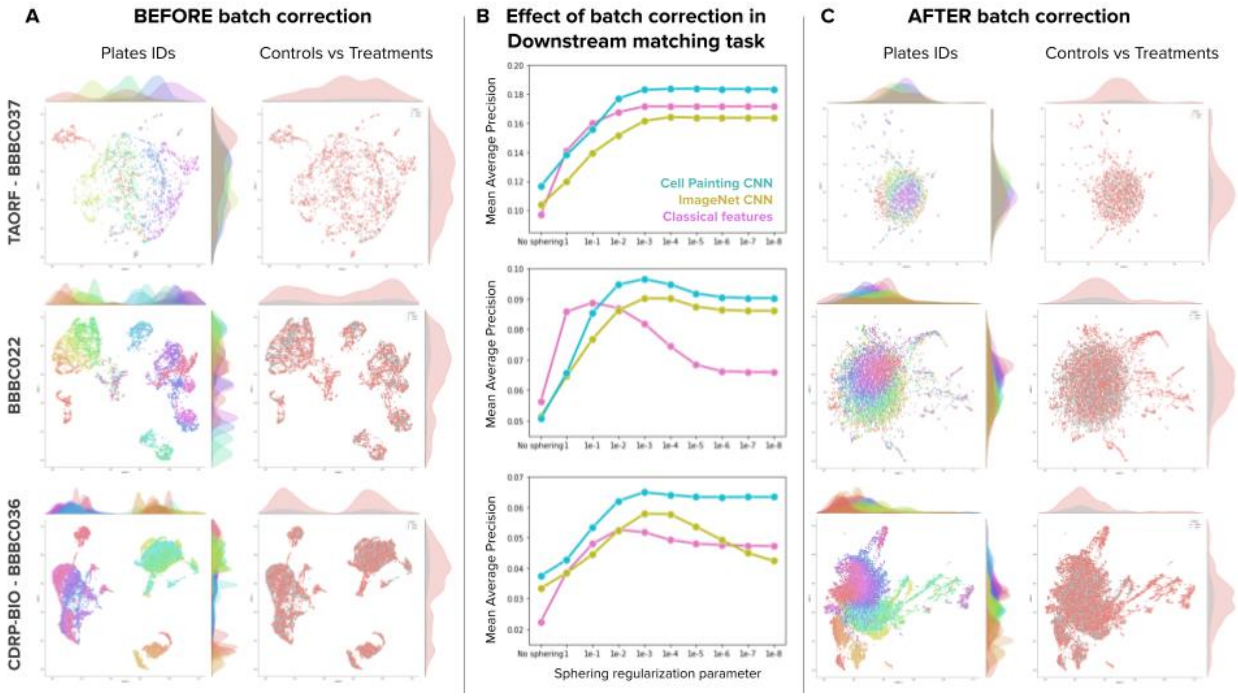


Figure 4. Effect of batch correction on feature representations. *Batch correction is based on the sphering transform and applied at the well-level, before treatment-level profiling (Methods). A) UMAP plots of well-level profiles before batch correction for the three benchmark datasets (rows) colored by plate IDs (left column) and by control vs treatment status (right column). The UMAP plots display density functions on the x and y axes for each color group to highlight the spread and clustering patterns of data. B) Effect of batch correction in the biological matching task. The x axis indicates the value of the regularization parameter of the sphering transform (smaller parameter means more regularization), with no correction in the leftmost point and then in decreasing order. The y axis is Mean Average Precision in the biological matching task. C) UMAP plots of well-level profiles after batch correction for the three benchmark datasets with the same color organization as in A.*

Discussion

In this study, we used three large Cell Painting benchmarks to evaluate training strategies for carrying out successful queries identifying matching compounds or genes using image-based profiling. Using a causal graph as a conceptual framework for analyzing the results, we found that WSL captures confounding and phenotypic variation in the latent variables. We also found that treatments with strong estimated average treatment effect have the potential to improve the quality of representations. Thus, we constructed a large training resource by combining five sources of Cell Painting data to maximize phenotypic and technical variation for training a reusable model for learning features. This model successfully improved performance in all the benchmarks and while also being computationally efficient.

The fact that the best-performing strategy involved training a single model once and profiling all the three benchmarks without re-training or fine-tuning has a remarkable implication: it indicates that generating large experimental datasets with a diversity of phenotypic impacts could be used to create a single model for the community that could be transformational in the same way that models trained on ImageNet have enabled transfer learning on natural image tasks. Private companies with large image sets might create and share such models. Alternatively, the upcoming public release of the JUMP-Cell Painting Consortium's dataset of more than 100,000 chemical and genetic perturbations, collected across 12 different laboratories in academia and industry, would be an excellent target for such an effort [35].

Methods

1. Datasets

1.1 Benchmarks and ground truth annotations

For this study, we employ five publicly available Cell Painting datasets: gene overexpression (BBBC037[34] and BBBC043 [11]) and compound screening (BBBC022 [31], BBBC036 [32] and LINCS [5]). BBBC037, BBBC022 and BBBC036 are U2OS cell-line datasets and were used for training and evaluation of profiling, while BBBC043 and LINCS are A549 cell-line and those were only partially used to construct a combined Cell Painting dataset (discussed further). Compounds in BBBC022, BBBC036 and LINCS partially overlap. In the compound screening datasets, DMSO was used as a negative control, in gene overexpression datasets genes were not perturbed in the negative control samples.

BBBC036 and BBBC037, BBBC022 did undergo quality control by analyzing the extracted features with principal component analysis. The outliers observed in the first two principal components were suspected to be candidates for exclusion. Those were visually inspected and found to be noisy or empty and not suitable for training and evaluation. With this quality control, two wells were removed from BBBC037 and 43 wells from BBBC022. Nothing was filtered from BBBC036.

If treatment was present in multiple concentrations in BBBC022 and BBBC036, we kept only the maximum concentration in the dataset for further training and evaluation.

The ground-truth annotations were reused from [34] with minor modifications (see Data Availability section). Only treatments with at least two replicates left after quality control are included in the ground-truth.

1.2 Selection of the strong treatments

The strongest treatments are selected using batch-corrected features obtained with CellProfiler (with sphering regularization parameter 1e-2) in the following way:

For each plate:

1. Calculate the median profile of the control subgroup of wells in the plate (median control profile)
2. For each treatment, calculate Euclidean distances between the treatment well-level feature vector and the median control profile feature vector.
3. Calculate Euclidean distances between the median control profile feature vector and each control well feature vector from the same plate.
4. Use the distances from the previous step to calculate the mean and standard deviation.
5. Using statistics obtained in step 4, apply Z-scoring to the distances obtained in step 2.

After obtaining the Z-scores for each plate, calculate the mean Z-score for each treatment. Rank treatments by mean Z-score.

1.3 Combined Cell Painting dataset

Motivated to train robust neural network models for feature extraction and an, we decided to take advantage of combining five publicly available Cell Painting datasets to maximize both phenotypic and technical variation. Treatments for the combined Cell Painting dataset are selected using the approach described in the previous section.

Technical variation is primarily encoded by control wells that are present in each plate of the experiment in several wells. Additionally, the negative controls are different between compound screening datasets and genetic perturbation datasets. To maximize the biological variation, first, we incorporate Cell Painting datasets of different cell-lines and also try to make an overlap of treatments between different cell lines to highlight the effect of the treatment. Second, we aimed to select the strongest phenotypes out of every dataset.

The selection started with BBBC022 and BBBC036 datasets. We selected the top 500 strongest treatments according to our approach from BBBC022 and took an overlap between BBBC022 and BBBC036, which resulted in 301 treatments. We additionally selected 50 from BBBC022 and 62 from BBBC036. Out of those 413 treatments, 122 overlapped with the LINCS dataset. We additionally selected 7 random treatments from LINCS, from top 20 (by number of associated treatments) MoAs.

Selection from BBBC037 and BBBC043 was similar, we selected 28 overlapping genes To increase overlapping we considered the selected “wildtype” genes as the same from different datasets, even though in BBBC037 the wildtypes were annotated to have different sub-populations. Additionally, we selected 29 top strongest perturbations from the BBBC037 dataset and the top strongest perturbations 32 from BBBC043 from non-overlapping subsets.

Negative controls from compound screening datasets and negative controls from gene overexpression datasets are considered as different classes in the combined dataset. Not all control wells were included from the LINCS dataset in the final dataset, as the control cells would dominate the final dataset, so we randomly sampled three control wells per plate.

As the final step, the treatments with less than 100 cells were filtered out.

In total the dataset contains 490 classes (488 for treatments and 2 for negative controls), 8 423 455 individual single-cells (47% treatment and 53% control cells), the diagrams are in Figure 1C.

2. Profiling workflow

2.1 Segmentation

The cell segmentation for all evaluated (BBBC037, BBBC022 and BBBC036) datasets is performed with methods built in CellProfiler v2 based on Otsu thresholding [36] and propagation method [37] based on Voronoi diagrams [38] or watershed from [39]. The segmentation is two-stepped: first, the images stained with Hoechst (DNA channel) were segmented using global Otsu thresholding. This prior information is then used in the second step: cell segmentation with the propagation or watershed method. The input channel for the second step depends on the dataset, as well as the other specific parameters of segmentation. The segmentation part of the pipelines is available in the published CellProfiler pipelines (see Code availability section).

2.2 Single-cell feature extraction with CellProfiler

Feature extraction for evaluated datasets was performed with CellProfiler 2. The feature extraction steps described in CellProfiler pipelines that are published together with used datasets. and can be grouped in several stages: 1) Data loading - load full image 2) Illumination correction for each channel 3) Identification of cell nuclei 4) Identification of cells using identified nuclei 4) Measurements: intensity, context, radial distribution, size and shape, texture 5) Export the features and cell outlines. Parameters of feature extraction can be found in CellProfiler pipelines which are available in published pipelines (see Code availability section).

2.3 CellProfiler features

The features of CellProfiler are designed to be human-readable and grouped into three large groups: “Cell”, “Cytoplasm” and “Nuclei”. Each of those feature groups has several common subgroups, such as shape features, intensity-based features, texture features and context features [12]. The resulting size of a feature vector is approximately 1800.
In this analysis, we reused well-level CellProfiler features from S3 buckets (see Data availability section) to generate the baseline results.

2.4 Feature aggregation

Profiles of single-cells are aggregated using median at field-of-view (image) level, fields-of-view are aggregated using mean to a well-level profile and, finally, wells are aggregated to treatment level profiles using mean aggregation. The feature aggregation steps are the same for CellProfiler and deep learning features. CellProfiler well-level features with NA values were filtered in the aggregation pipeline.

2.5 Batch correction using sphering transform

One of the possible post-processing methods to reduce batch effects and remove other possible nuisance variations is Typical Variation Normalization (TVN) [18]. Axes of variation are computed using Principal Component Analysis (PCA), then the axes with little variation are amplified and the importance of axes with large variation is reduced by axis normalization. We can control the strength of signal amplification\reduction by using a regularization parameter. The computation at this point involves only profiles of negative controls and as a result, we get the TVN transformation. Then, this transformation is applied to all well-level feature vectors in the analyzed dataset.

Sphering transformation used in this paper, similarly it n well-level profiles of negative controls with vector size d as an input $X^{n \times d}$. Then covariance matrix is calculated $\Sigma = \frac{X^T X}{n}$ and its eigendecomposition is $Q = U \Delta U^T$, where Δ are eigenvalues. Then we divide the diagonal by the square root of eigenvalues in addition with a regularization parameter λ. The final ZCA-transformation [40] matrix is $U\,(\Delta + \lambda)^{-1} U^T$.

The effect of sphering and its regularization on representations and profiling performance is demonstrated in Figure 4.

2.6 Similarity matching

To assess the similarity between resulting treatment profiles cosine similarity is measured for each pair of treatments. Cosine similarity is one of several similarity metrics [12] in profiling and used in [33].

$$\text{cosine similarity} = \frac{A \cdot B}{||A||\,||B||}$$

3. Deep learning models

3.1 EfficientNet

All deep learning experiments were conducted with EfficientNet [28]. We use the base model EfficientNet-B0 as the size of single-cell crops is only 128x128. It consists of 9 stages: input, seven inverted convolutional blocks from MobileNetV2 [41] (with the addition of squeeze and excitation optimization) and final layers. The usage of convolutional blocks from MobileNetV2 in combination with neural architecture search gave EfficientNet an advantage in terms of computational efficiency and accuracy compared to ResNet50.

EfficientNet was used in biological tasks, for instance, there is an effort to create a model, pre-trained on cellular images called CytoImageNet [42]. Pre-trained EfficientNet was used to extract single-cell embeddings [11] of A549 cells. EfficientNets were also used in cell classification Kaggle competition of Recursion Pharmaceuticals (https://www.kaggle.com/competitions/recursion-cellular-image-classification/), the highest scoring solution based on EfficientNet took 6th place in the private leaderboard.

3.2 Image preprocessing

The raw Cell Painting images are 16-bit TIFF format. To ease the processing for deep learning models, we perform image compression and normalization. The normalization for each image is the following:

- 99.99 percentile of the plate's illumination distribution.
- Compute 0.05 and 99.95 percentiles of illumination distribution in the current image.
- Clip illumination intensity using 0.05 percentile and the minimum of 99.99 percentile of the plate's illumination distribution and 99.95 percentile of illumination distribution in the current image.
- Rescale intensity, create a histogram with 256 bins and save as PNG.

This preprocessing pipeline is included in DeepProfiler and can be run with metadata which lists all the images in the dataset, together with plate, well and site (image or field of view) information.

3.3 Feature extraction with ImageNet pretrained models

One approach adopted to extract representations is to use pre-trained models on a dataset of similar or different domains. Most of the popular existing deep learning architectures are pre-trained on the ImageNet dataset [43], which is composed of natural images.
The idea to use such pre-trained models for morphological profiling starts in the experiments demonstrated in [17] with VGG-16, Inception v3 and ResNet models on the BBBC021 dataset. A similar approach was used in a subset of the lung adenocarcinoma dataset [20] and then in full with pre-trained EfficientNet-B0 [11]. On a large scale, pre-trained models were also used [10], though the dimensionality of extracted features was reduced.

We use DeepProfiler to extract features of single-cells with the pre-trained EfficientNet. The input single-cell images of size 128x128 are cropped out from full images. To fit the expected input size of 224x224, the crops are resized. The pixel values are then rescaled using min-max normalization and adjusted to have values [-1:1] to match the required input range.
As Cell Painting images are five-channel and ImageNet pre-trained models are for three-channel (RGB) images, additional changes are required. For that, we simply replicate each channel three times and pass through the model (so each cell requires five inference passes). Features extracted for each channel are concatenated and the resulting feature vector size is 3360 (672 is the size of the block6a_activation layer in the used EfficientNet implementation).

3.4 Weakly Supervised Learning

Weakly supervised learning (WSL) in this analysis means that we train the models not to classify single-cells by mechanism of action or pathway (that we don't know in a real-world setting), but with auxiliary task of single-cell classification by treatments (in a real-world setting we always know the treatments). Even though we know the treatments, the data is noisy due to several reasons [20]: 1) treatments might not significantly affect the phenotype of the cell (the treatment is neutral or technical problems occurred in the experiment) 2) Applying different treatments might

yield cells with similar phenotype 3) Cells might not react uniformly to particular treatments [44], which yields not necessarily meaningful subpopulations of cells.

3.5 Training of Cell Painting models

The Cell Painting models are trained on crops of single-cells, those are obtained from full images using cell locations and full-image metadata. To gather the cropped version of the dataset, we use DeepProfiler, it saves five unfolded channels as a single stripe and generates metadata for single-cells. The stripe can also contain a single-cell segmentation mask so it can be used to filter the crop from neighboring cells and noise (cell masking is not used in the reported analysis). The crop size can be determined by the user and depends on the dataset.
DeepProfiler performs rebalancing of the data for each epoch of training to have an approximately equal number of training examples for each target class (treatment). That is especially important to prevent the overrepresentation of controls and a few treatments during training.
The data augmentation pipeline in DeepProfiler consists of three steps:

1. Random crop and resize. This augmentation is applied with 0.5 probability. The crop region size is random, 80% to 100% of the size of the original image, then resized back to the original size.
2. Random horizontal flips and 90-degree rotations.
3. Brightness and contrast adjustments with built-in Tensorflow methods, each channel is adjusted separately for both augmentation steps.

The parameters for training in all experiments were the following: single-cell crop size 128x128, SGD optimizer, learning rate 0.005, batch size 32, augmentations on, no label smoothing or online label smoothing, 30 epochs of training, the results for all experiments are reported for the last epoch. The model is initialized with ImageNet weights. All models were trained with categorical cross-entropy loss.
We tested two data-split approaches for train and validation subsets for the treatment classification task:

- Leave plates out: we selected the plates in a way that the data of one subset of plates is only the train data and another one is in validation.
- Leave cells out: in all plates present in the training and validation set, we split the data randomly on the well-level, meaning that approximately 60% of cells from each well would be in the training set and the rest in the validation set.

The training of deep learning models was performed on NVIDIA DGX with NVIDIA V100 GPUs; a single GPU was used to train each model.

3.6 Feature extraction with trained Cell Painting models

We use DeepProfiler to extract features, profiling yields one NumPy file (one per image) with an array of vectors (one per cell). DeepProfiler uses full image metadata and location files for feature extraction. As the model is now natively trained for five-channel images, no need to replicate each channel separately, thus, each cell requires one inference pass and the feature vector contains

representation of all channels simultaneously. The size of the feature vectors is 672 for reported results, which were extracted from EfficinetNet's block6a_activation layer.

3.7 DeepProfiler

We developed DeepProfiler, a tool for learning and extracting representations from high-throughput microscopy images using convolutional neural networks (CNNs). DeepProfiler proposes a standardized workflow which unifies image pre-processing, training of CNNs and feature extraction discussed in the previous sections. DeepProfiler is implemented in Tensorflow [45] (version 2), the code is publicly available on GitHub (see Code Availability).

4. Evaluation

The evaluation task mostly replicates the one from [34], we check if the most similar treatments above a certain threshold do share the same gene pathway or mechanism of action (MoA). For that, we use modified folds of enrichment metric (based on [34]), in addition we adopt mean average precision and propose a new metric: top hits in the top 1%, which can be measured at different levels of profiling. Features of single-cell are extracted and used for evaluation disregarding their belonging to the training or validation subset of the pretext task.
For folds of enrichment, precision-recall metrics and first-hit on treatment-level we define query and the response as follows:

- Query treatments - the treatment which belongs to an MoA\pathway with at least two treatments, thus it is possible to find a match.
- Response treatments - all the treatments, but the query treatment. Response treatments include the treatments belonging to MoAs\pathways with one treatment.

4.1 Folds of enrichment

For each query treatment we calculate odds ratio in a one-sided Fisher's exact test. The test is calculated using a 2x2 contingency table: the first row contains a number of treatments with the same MoAs\pathways (matches) and different MoAs\pathways at a selected threshold, the second row is the same, but beyond the threshold. Odds ratio is a sum of the first row divided by the sum of the second row. It demonstrates the likelihood of observing the treatment with the same MoA\pathway in the top connections.
We report the mean odds ratio over all query treatments. The threshold we use is 1% of connections. This metric in the text is referred to as "Folds of Enrichment".
The implementation of the metric is available as a part of analysis pipelines (see Code availability section).

4.2 Mean Average Precision

For each query treatment average precision (area under precision-recall curve) is computed for the information retrieval task. The search starts from the most similar treatments to the query and

continues until all positive pairs (response treatments with the same MoA\pathway) are found, precision and recall are computed at each step of the search.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

where TP - true positive, FP - false positive, FN - false negative. Interpolated precision at recall r As the number of treatments per MoA\pathway is not balanced, the precision-recall curve has a different number of recall points; precision and recall are interpolated to cover the maximum number of recall points possible in the dataset. Interpolated precision at each recall point is defined as follows [46]:

$$p_{inter}(r) = max_{r' \geq r}\, p(r')$$

Average precision for query treatment is the mean of p_{inter} at all recall points. The reported mean average precision (mAP) is the mean average precision over all queries.

4.3 Hits in the top 1%

The purpose of computational drug discovery is to select a small portion of candidate treatments from the full set of treatments in physical lab experiments. To simulate this, we introduce ***hits in the top 1%*** metric, which measures the number of query treatments which have a treatment of the same MoA/pathway (a hit) in the top 1% of responses (responses are ranked by similarity measure).

This metric is applicable for several levels of profiling: we report results for image level, well level and treatment level. For image-level and well-level we search for an image or well of the same treatment as the treatment of the query image or well. In case of image-level the images belonging to the same well as the query image, are excluded from possible responses. In the well-level analysis, the query well is also removed from possible responses. All treatments in the dataset can be queries on those two levels of profiling.

Acknowledgements

We thank Salil Bhate for the valuable discussions and feedback provided throughout the development of this project.

NM acknowledges the short-term scientific mission grants provided by eCOST action CA15124 (NEUBIAS) in 2019 and 2020. NM and PH acknowledge support from the LENDULET-BIOMAG Grant (2018-342), from the European Regional Development Funds (GINOP-2.2.1-15-2017-00072), from the H2020 and EU-Horizont (ERAPERMED-COMPASS, ERAPERMED-SYMMETRY, DiscovAIR, FAIR-CHARM) from TKP2021-EGA09, from the ELKH-Excellence grants and from the Cooperative Doctoral Programme (2020-2021) of the Ministry for Innovation and Technology. Researchers in the Carpenter–Singh lab were supported by NIH R35 GM122547 to AEC. Researchers in the Cimini lab were supported by NIH P41 GM135019. JCC was supported by the Schmidt Fellowship program of the Broad Institute and by the NSF Award 2134695.

Data availability

Ground truth files for evaluation are available on GitHub https://github.com/broadinstitute/DeepProfilerExperiments.
Link to the pre-trained EfficientNet model https://github.com/Callidior/keras-applications/releases/download/efficientnet/efficientnet-b0_weights_tf_dim_ordering_tf_kernels_autoaugment.h5

The Cell Painting datasets (raw images and CellProfiler profiles) are available at public S3 buckets:
BBBC037 gene overexpression dataset in U2OS cells [30]
s3://cytodata/datasets/TA-ORF-BBBC037-Rohban/profiles_cp/TA-ORF-BBBC037-Rohban/

BBBC022 compound screening in U2OS cells [31]
s3://cytodata/datasets/Bioactives-BBBC022-Gustafsdottir/profiles/Bioactives-BBBC022-Gustafsdottir/

BBBC036 compound screening in U2OS cells [32] s3://cytodata/datasets/CDRPBIO-BBBC036-Bray/profiles_cp/CDRPBIO-BBBC036-Bray/

BBBC043 gene overexpression dataset in A549 cells [11]
s3://cytodata/datasets/LUAD-BBBC043-Caicedo/profiles_cp/LUAD-BBBC043-Caicedo/

LINCS compound screening in A549 cells [5]
s3://cellpainting-gallery/cpg0004-lincs/broad/images/2016_04_01_a549_48hr_batch1/

Code availability

The DeepProfiler code is available on GitHub https://github.com/cytomining/DeepProfiler.
The analysis pipelines (Jupyter notebooks and Python scripts to analyze features) are available on GitHub https://github.com/broadinstitute/DeepProfilerExperiments.
DeepProfiler documentation is available here: https://cytomining.github.io/DeepProfiler-handbook/
In DeepProfiler we used the following EfficientNet implementation: https://github.com/qubvel/efficientnet.

The code and CellProfiler pipelines for three evaluated datasets can be found in the associated GitHub repositories:
BBBC037: https://github.com/carpenterlab/2017_rohban_elife
BBBC036: https://github.com/gigascience/paper-bray2017
BBBC022: Supplementary materials in [31].

References

1. Bray, M.-A. *et al.* Cell Painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nat. Protoc.* **11**, 1757–1774 (2016).
2. Cimini, B. A. *et al.* Optimizing the Cell Painting assay for image-based profiling. *bioRxiv* 2022.07.13.499171 (2022) doi:10.1101/2022.07.13.499171.
3. Wawer, M. J. *et al.* Toward performance-diverse small-molecule libraries for cell-based phenotypic screening using multiplexed high-dimensional profiling. *Proceedings of the National Academy of Sciences* **111**, 10911–10916 (2014).
4. Cuccarese, M. F. *et al.* Functional immune mapping with deep-learning enabled phenomics applied to immunomodulatory and COVID-19 drug discovery. 2020.08.02.233064 (2020) doi:10.1101/2020.08.02.233064.
5. Way, G. P. *et al.* Morphology and gene expression profiling provide complementary information for mapping cell state. *bioRxiv* 2021.10.21.465335 (2021) doi:10.1101/2021.10.21.465335.
6. Simm, J. *et al.* Repurposing High-Throughput Image Assays Enables Biological Activity Prediction for Drug Discovery. *Cell Chem Biol* **25**, 611–618.e3 (2018).
7. Way, G. P. *et al.* Predicting cell health phenotypes using image-based morphology profiling. *Mol. Biol. Cell* mbcE20120784 (2021).
8. Moshkov, N. *et al.* Predicting compound activity from phenotypic profiles and chemical structures. *bioRxiv* 2020.12.15.422887 (2022) doi:10.1101/2020.12.15.422887.
9. Rohban, M. H. *et al.* Virtual screening for small molecule pathway regulators by image profile matching. *bioRxiv* 2021.07.29.454377 (2022) doi:10.1101/2021.07.29.454377.
10. Schiff, L. *et al.* Integrating deep learning and unbiased automated high-content screening to identify complex disease signatures in human fibroblasts. *bioRxiv* 2020.11.13.380576 (2021) doi:10.1101/2020.11.13.380576.

11. Caicedo, J. C., Arevalo, J. & Piccioni, F. Cell Painting predicts impact of lung cancer variants. *Mol. Biol. Cell* (2022).

12. Caicedo, J. C. *et al.* Data-analysis strategies for image-based cell profiling. *Nat. Methods* **14**, 849–863 (2017).

13. Caicedo, J. C., Singh, S. & Carpenter, A. E. Applications in image-based profiling of perturbations. *Curr. Opin. Biotechnol.* **39**, 134–142 (2016).

14. Chandrasekaran, S. N., Ceulemans, H., Boyd, J. D. & Carpenter, A. E. Image-based profiling for drug discovery: due for a machine-learning upgrade? *Nat. Rev. Drug Discov.* (2020) doi:10.1038/s41573-020-00117-w.

15. McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biol.* **16**, e2005970 (2018).

16. Stirling, D. R. *et al.* CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics* **22**, 433 (2021).

17. Pawlowski, N., Caicedo, J. C., Singh, S., Carpenter, A. E. & Storkey, A. Automating Morphological Profiling with Generic Deep Convolutional Networks. *bioRxiv* 085118 (2016) doi:10.1101/085118.

18. Michael Ando, D., McLean, C. Y. & Berndl, M. Improving Phenotypic Measurements in High-Content Imaging Screens. *bioRxiv* 161422 (2017) doi:10.1101/161422.

19. Schiff, L. *et al.* Integrating deep learning and unbiased automated high-content screening to identify complex disease signatures in human fibroblasts. *Nat. Commun.* **13**, 1590 (2022).

20. Caicedo, J. C., McQuin, C., Goodman, A., Singh, S. & Carpenter, A. E. Weakly Supervised Learning of Single-Cell Feature Embeddings. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2018**, 9309–9318 (2018).

21. Lu, A. X., Kraus, O. Z., Cooper, S. & Moses, A. M. Learning unsupervised feature representations for single cell microscopy images with paired cell inpainting. *PLoS Comput.*

Biol. **15**, e1007348 (2019).

22. Hofmarcher, M., Rumetshofer, E. & Clevert, D. A. Accurate prediction of biological assays with high-throughput microscopy images and convolutional networks. *Journal of chemical* (2019).
23. Yang, S. J. *et al.* Applying Deep Neural Network Analysis to High-Content Image-Based Assays. *SLAS Discov* **24**, 829–841 (2019).
24. Mao, C. *et al.* Generative Interventions for Causal Learning. in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021). doi:10.1109/cvpr46437.2021.00394.
25. Schölkopf, B. *et al.* Toward Causal Representation Learning. *Proc. IEEE* **109**, 612–634 (2021).
26. Rubin, D. B. Estimating causal effects of treatments in randomized and nonrandomized studies. *J. Educ. Psychol.* **66**, 688–701 (1974).
27. Johansson, F., Shalit, U. & Sontag, D. Learning Representations for Counterfactual Inference. in *Proceedings of The 33rd International Conference on Machine Learning* (eds. Balcan, M. F. & Weinberger, K. Q.) vol. 48 3020–3029 (PMLR, 20--22 Jun 2016).
28. Tan, M. & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv [cs.LG]* (2019).
29. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* (2018) doi:10.1038/nbt.4314.
30. Rohban, M. H. *et al.* Systematic morphological profiling of human gene and allele function via Cell Painting. *Elife* **6**, (2017).
31. Gustafsdottir, S. M. *et al.* Multiplex cytological profiling assay to measure diverse cellular states. *PLoS One* **8**, e80999 (2013).
32. Bray, M.-A. *et al.* A dataset of images and morphological profiles of 30 000 small-molecule treatments using the Cell Painting assay. *Gigascience* **6**, 1–5 (2017).

33. Ljosa, V. *et al.* Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. *J. Biomol. Screen.* **18**, 1321–1329 (2013).

34. Rohban, M. H., Abbasi, H. S., Singh, S. & Carpenter, A. E. Capturing single-cell heterogeneity via data fusion improves image-based profiling. *Nat. Commun.* **10**, 2082 (2019).

35. Chandrasekaran, S. N. *et al.* Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *bioRxiv* 2022.01.05.475090 (2022) doi:10.1101/2022.01.05.475090.

36. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979).

37. Jones, T. R., Carpenter, A. & Golland, P. Voronoi-Based Segmentation of Cells on Image Manifolds. in *Computer Vision for Biomedical Image Applications* 535–543 (Springer Berlin Heidelberg, 2005).

38. Voronoi, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *J. Reine Angew. Math.* **1908**, 198–287 (1908).

39. van der Walt, S. *et al.* scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014).

40. Kessy, A., Lewin, A. & Strimmer, K. Optimal Whitening and Decorrelation. *Am. Stat.* **72**, 309–314 (2018).

41. Sandler, Howard & Zhu. Mobilenetv2: Inverted residuals and linear bottlenecks. *Proc. Estonian Acad. Sci. Biol. Ecol.*

42. Hua, S. B. Z., Lu, A. X. & Moses, A. M. CytoImageNet: A large-scale pretraining dataset for bioimage transfer learning. *arXiv [cs.CV]* (2021).

43. Deng, J. *et al.* ImageNet: A large-scale hierarchical image database. in *2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255 (2009).

44. Gough, A. *et al.* Biologically Relevant Heterogeneity: Metrics and Practical Insights. *SLAS*

Discov **22**, 213–237 (2017).

45. Developers, T. *TensorFlow*. (Zenodo, 2021). doi:10.5281/ZENODO.4724125.

46. Manning, C. D. *Introduction to information retrieval*. (Syngress Publishing, 2008).