RxRx1: A Dataset for Evaluating Experimental Batch Correction Methods

Maciej Sypetkowski[1] Morteza Rezanejad[1] Saber Saberian[1] Oren Kraus[1]
John Urbanik[1] James Taylor[2] Ben Mabey[1] Mason Victors[1] Jason Yosinski[3]
Alborz Rezazadeh Sereshkeh[1] Imran Haque[1] Berton Earnshaw[1]
[1]*Recursion* [2]*Enveda Biosciences* [3]*ML Collective*

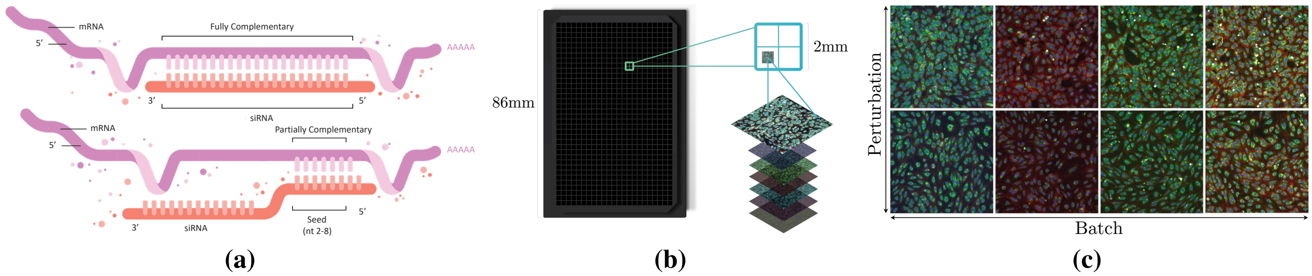


Figure 1. **(a)** Top: Depiction of full-complementarity of an siRNA to an mRNA to knockdown a particular target gene. Bottom: Depiction of partial-complementarity in the seed-region of an siRNA, leading to partial knockdown of hundreds of additional genes. **(b)** Schematic of a 384-well plate demonstrating imaging sites and 6-channel images. The 4-plate experiments in the dataset were run in such plates. RxRx1 contains two 6-channel images from different sites per well. **(c)** Images of two different genetic conditions (rows) in HUVEC cells across four experimental batches (columns). Notice the visual similarity of images from the same batch.

Abstract

High-throughput screening techniques are commonly used to obtain large quantities of data in many fields of biology. It is well known that artifacts arising from variability in the technical execution of different experimental batches within such screens confound these observations, and can lead to invalid biological conclusions. It is, therefore, necessary to account for these batch effects *when analyzing outcomes. In this paper, we describe* RxRx1*, a biological dataset designed specifically for the systematic study of batch effect correction methods. The dataset consists of 125,510 high-resolution fluorescence microscopy images of human cells under 1,138 genetic perturbations in 51 experimental batches across 4 cell types. Visual inspection of the images clearly demonstrates significant batch effects. We also propose a classification task designed to evaluate the effectiveness of experimental batch correction methods on these images and examine the performance of a number of correction methods on this task. Our goal in releasing RxRx1 is to encourage the development of effective experimental batch correction methods that generalize well to unseen experimental batches. The dataset can be downloaded at `https://rxrx.ai`.*

1. Introduction

High-throughput screening is commonly used in many biological fields, including genetics [18, 53] and drug discovery [5, 7, 36, 49]. Such screens are capable of generating large amounts of data that, when coupled with modern machine learning methods, could help answer fundamental questions in biology and solve the problem of rising costs in drug discovery, which are now estimated to be well over 2 billion per approved drug [16, 44]. However, creating large volumes of biological data necessarily requires the data to be generated in multiple experimental batches, or groups of experiments executed at similar times under similar conditions. Even when experiments are carefully designed to control for technical variables such as temperature, humidity, and reagent concentration, the measurements taken from these screens are confounded by artifacts that arise from differences in the technical execution of each batch. Figure 1c demonstrates the complexity of identifying relevant biological variation and separating it from technical noise caused by these so-called *batch effects*. Batch effects can alter factors of variation within the images that are irrelevant to the biological variables under study, but unfortunately are often correlated with them. It is, therefore, necessary to correct for such effects before drawing any bi-

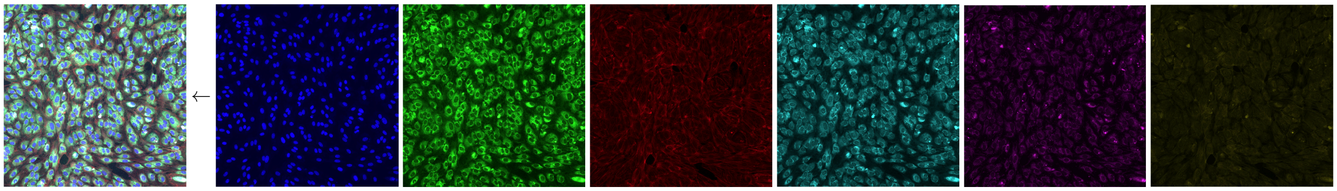


Figure 2. 6-channel faux-colored composite image of HUVEC cells (left) and individual channels (right) stained with Hoechst 33342 (channel 1, blue), Alexa Fluor 488 Concanavalin A (channel 2, green), Alexa Fluor 568 Phalloidin (channel 3, red), Syto14 (channel 4, cyan), MitoTracker Deep Red FM (channel 5, magenta), Alexa Fluor 555 Agglutinin (channel 6, yellow). The similarity in content between some channels is due in part to the spectral overlap between the fluorescent stains used in those channels.

ological conclusions [4, 26, 30, 40, 41, 48]. Indeed, many computational methods have been designed for correcting such effects [21–23, 27, 31, 34, 35, 45].

In this paper, we describe the *RxRx1* dataset, an image dataset systematically designed to study batch effect correction methods. The dataset consists of 125,510 6-channel fluorescence microscopy images of human cells under 1,108 different genetic perturbations (plus 30 positive control perturbations) across 51 experimental batches and 4 cell types. We propose a machine learning task to gauge the effectiveness of batch effect correction methods, namely learning to classify the genetic perturbation present in each image in a set of experimental batches held out from a training set. In order for a classifier to perform well on this task, it must be able to robustly identify the discriminative morphological features associated with each genetic perturbation against a background of the latent technical variations associated with each held-out experimental batch.

In the present article, we make three main contributions:

1. We present a dataset (46GB, 125,510 images, 1,139 classes) for testing experimental batch effect correction, comparable in size to reference datasets such as ImageNet [15] (155 GB, 1.2M images, 1000 classes) and other biological datasets like BBBC017 (56 GB, 64.5K images, 4903 classes).

2. We introduce a specific task for evaluating the effectiveness of batch effect correction methods, accompanied by three evaluation metrics enabling users of this dataset to assess their developed methods.

3. We demonstrate the use of a standard convolutional classifier architecture as a backbone for the task of experimental batch correction and analyze the performance of variations of this model on such a task.

This dataset and task will be of interest to the community of researchers applying machine learning methods to complex biological datasets, especially those working with image-based high-content phenotypic screens [1, 2, 8, 11, 28, 29]. In addition, we hope RxRx1 is of interest to the larger community of machine learning researchers working in the areas of domain adaptation, transfer learning, and few-shot learning.

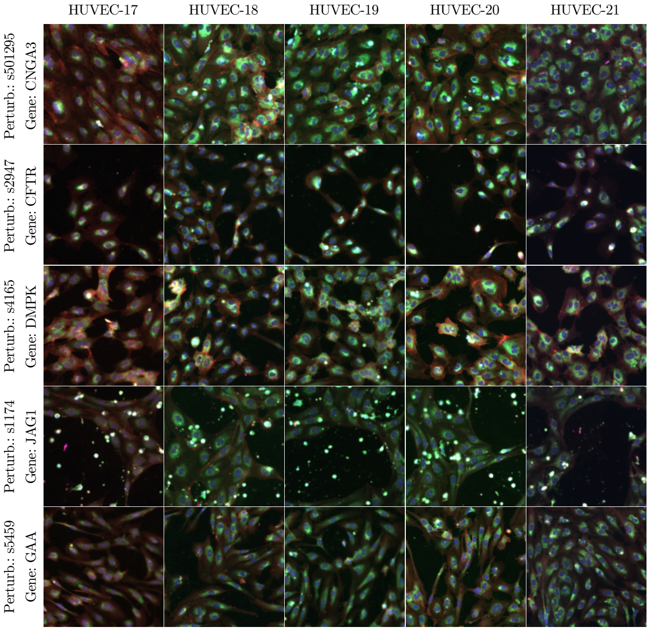


Figure 3. Images of 5 siRNA phenotypes in HUVEC cells across 5 experimental batches. Each siRNA causes changes in the visual properties of cell populations, including morphology, count, and spatial distribution.

2. Dataset

All images in RxRx1 were generated using Recursion's high-throughput screening platform[1]. The dataset is comprised of fluorescence microscopy images of human cells of four different types:

- HUVEC: primary endothelial cells derived from the umbilical vein [14].

- RPE: epithelial cells from the outermost layer of the retina [51].

[1] https://recursion.com

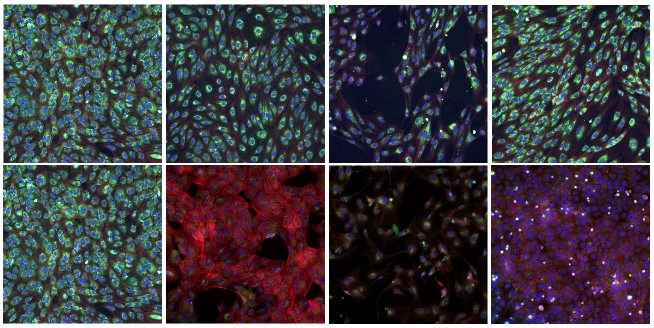


Figure 4. Top row: Images of HUVEC cells under four different siRNA perturbations, all from the same plate. Bottom row: Images of cells under the same siRNA perturbation in four cell types: HUVEC, RPE, HepG2, and U2OS.

- HepG2: nontumorigenic cells with high proliferation rates and an epithelial-like morphology important for hepatic functions [17].

- U2OS: immortalized epithelial cells derived in 1964 from an osteosarcoma patient [39].

These were acquired using a proprietary implementation of the Cell Painting imaging protocol [6]. In Figure 2, we show an example image. Each channel corresponds to a fluorescent dye used to stain one of six different targeted cellular components, namely the nucleus, endoplasmic reticulum, actin, nucleoli, mitochondria, and Golgi.

The images themselves are the result of executing the same experimental design in 51 different experimental batches, with each execution separated by at least a week from all others. The experiment design consists of four 384-well plates (see Figure 1b), where each well contains an isolated population of cells. The wells are laid out on each plate in a 16×24 grid, but only the wells in the inner 14×22 grid are used since the outer wells are most susceptible to environmental factors. In each well, cell populations are genetically perturbed with small interfering ribonucleic acid, or siRNA, at a fixed concentration. Each siRNA is designed to knockdown a single target gene via the RNA interference pathway, reducing the expression of the target gene [50]. In addition, siRNAs are known to have significant but consistent off-target effects via the microRNA pathway, creating partial knockdown of many other genes as well (see Figure 1a). Each siRNA, therefore, perturbs cellular function in a way that can impact visible properties of the cell population, including morphology, count, and spatial distribution (see Figure 3). The set of consistent, observable characteristics associated with each siRNA is called its *phenotype*. Note that the phenotype of an siRNA is sometimes visually distinct, but more often its visual characteristics are subtle and hard to detect by the eye (see Figure 4).

2.1. Experiment design

Of the 308 usable wells on each plate, one is left untreated to provide a negative control (labeled EMPTY), and another 30 wells receive a unique siRNA from a positive control set of 30 siRNA. The remaining 277 wells receive a unique siRNA from a treatment set of 1,108 siRNA. Therefore, each 4-plate experiment contains 1,138 unique siRNA perturbations, where the positive and negative controls appear once on each plate, and the 1,108 treatments appear once in each 4-plate experiment. The location of each siRNA is randomized per experiment and plate, though for operational reasons, the 1,108 treatment siRNA are divided into four groups of 277 that always appear together on a plate. Note that some wells do not receive their intended siRNA (and are thus labeled EMPTY) due to operational errors, while the images of other wells are removed from the dataset due to poor data quality.

2.2. Image resolution

Images were acquired at a spatial resolution of 2048 × 2048 and 16 bits per pixel per channel, downsampled to 1024 × 1024 at 8bpp and cropped to the center 512 × 512 field of view. RxRx1 contains two non-overlapping 512 × 512 fields of view per well. Of the possible 125,564 total images (51 experiments × 4 plates/experiment × 308 wells/plate × 2 images/well), 154 images were excluded for failing quality filters, resulting in a total of 125,510 6-channel images in the dataset.

2.3. Cell types

The 51 experiments are distributed across four cell types: 24 in HUVEC, 11 in RPE, 11 in HepG2, and 5 in U2OS. Figure 4 shows the phenotype of a single siRNA in the four different cell types. Each of the 51 experiments was run in a different batch, resulting in images that exhibit distinct batch effects. It is this feature of the dataset that makes it particularly suited for studying batch effects and methods for correcting them.

2.4. Metadata

The following metadata is provided for each image in RxRx1: cell type, experiment id, plate id, well location, and treatment class (1,138 siRNA classes plus one untreated class).

3. Evaluation task

We propose the following task for evaluating the effectiveness of batch effect correction methods: learn to classify the genetic perturbation present in each image in a set of experimental batches held out from a training set. In order for a classifier to perform well on this task, it must be able to robustly identify the visual characteristics associated

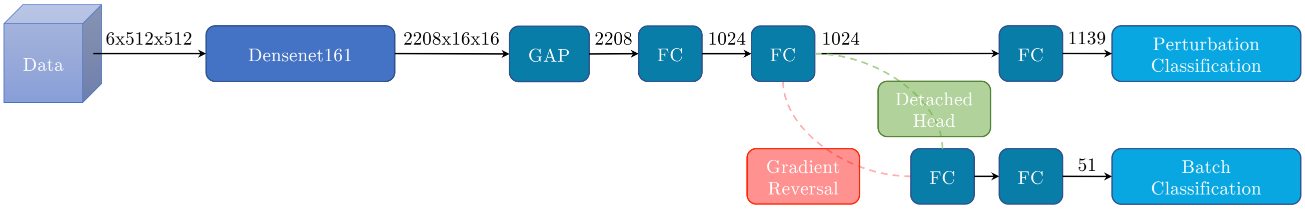


Figure 5. A diagram of our models. 6-channel images are fed to the backbone (DenseNet161 [24]). Feature maps from the backbone are pooled by global average pooling and then mapped by two fully connected layers, which follow batch normalization and ReLU layers to obtain a 1024-dimensional image embedding. The embedding layer is connected to two parallel branches – one for perturbation classification and the other for experimental batch classification. The experimental batch classification branch is either detached (for baseline and AdaBN model) or gradient reversed (for gradient reversal model). For both classification targets, we use cross-entropy loss.

with each genetic perturbation against a background of latent technical variations associated with each experimental batch.

3.1. Batch-separated vs batch-stratified splits

In order to appropriately evaluate such classifiers, we propose two ways of splitting RxRx1 into training and test sets. The first, called the *batch-separated* split, assigns 33 of the 51 experiments (16 HUVEC, 7 RPE, 7 HepG2, 3 U2OS) to the training set, and the remaining 18 (8 HUVEC, 5 RPE, 5 HepG2, 2 U2OS) to the test set. In this way, the experimental batches that make up the test set are different from those in the training set, which allows for assessing out-of-domain generalization. Note that this split is naturally stratified with respect to treatment class (see Section 2.1). The second split, called the *batch-stratified* split, stratifies the data by both treatment class and experimental batch. The size of the training and test sets are made roughly the same as in the batch-separated split. In the batch-stratified split, the training and test sets contain images from all experimental batches, making the classification task easier to learn since no experimental batch is out-of-domain. As a result, accuracy on the batch-stratified split sets an upper bound for accuracy on the batch-separated split, and we will use both of these numbers when evaluating experimental batch correction methods.

3.2. Evaluation metrics

With the batch-separated and batch-stratified splits defined, we now propose three evaluation metrics for assessing the effectiveness of experimental batch correction methods.

3.2.1 Perturbation classification accuracy

This metric is the average perturbation class classification accuracy (including controls and untreated as classes) on the test set when using the batch-separated split. It is useful as an overall measure of the goodness of the batch effect correction method since 1) the test set contains experimental batches not seen during training, 2) the training and test sets are stratified by siRNA classes, and 3) the metric improves with each correctly classified image.

3.2.2 Batch generalization

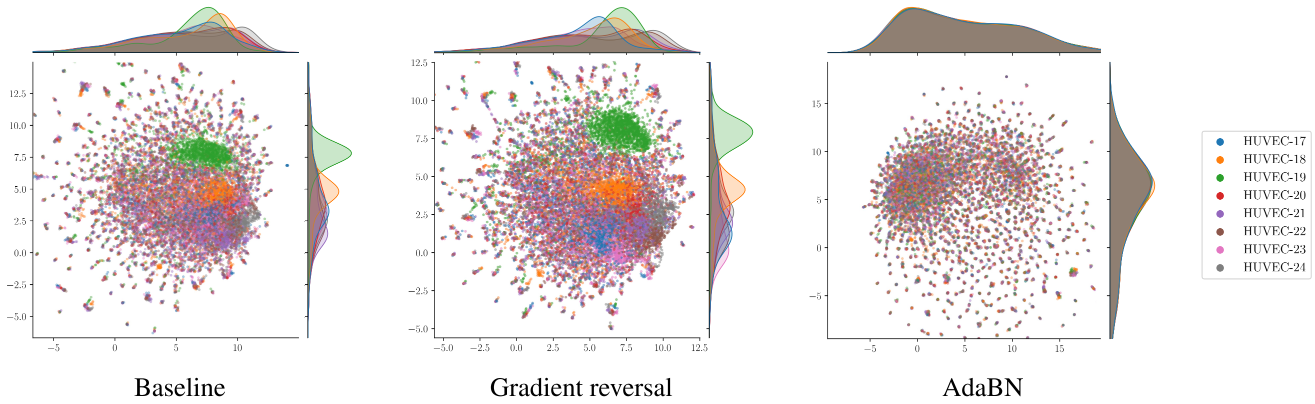
To define a metric that measures generalization to new experimental batches, we calculate perturbation classification accuracy using both the batch-separated and batch-stratified splits, and then measure the difference between these accuracies as follows:

$$Generalization = \frac{SeparatedPertAcc}{StratifiedPertAcc}$$

where *SeparatedPertAcc* is perturbation classification accuracy on the test set of the batch-separated split (after training on the batch-separated split), and *StratifiedPertAcc* is perturbation classification accuracy on the test set of the batch-stratified split (after training on batch-stratified split). A generalization of 100% means that perturbation classification accuracy on both splits is the same, *i.e*., the experimental batch correction method has learned to classify perturbations in unseen experimental batches as well as it has learned to classify perturbations in seen experiment batches.

3.2.3 Batch classification accuracy

To measure the content of experimental batch-related information encoded into the image embeddings, we calculate experimental batch classification on the batch-stratified split (since the training must contain experiment batches from the test set for this metric to make sense). Because experimental batch related information is an artifact, we want embeddings to be as invariant to batch as possible. Therefore, batch classification accuracy should be as close to random as possible, *i.e*., $1/51 \approx 1.96\%$.



Baseline Gradient reversal AdaBN

Figure 6. UMAP visualization of embedding spaces for baseline, gradient reversal, and AdaBN methods (AdaBN + gradient reversal UMAP is similar to AdaBN UMAP). Points represent embeddings of individual images in HUVEC experiment batches from the test set and are colored by the experimental batch (other cell types exhibit similar behavior). Note that while gradient reversal is able to reduce experimental batch classification accuracy to random when trained on the batch-stratified split, this behavior does not generalize well to unseen experimental batches. In contrast, AdaBN is far more effective in aligning unseen experiment batches.

4. Experimental batch correction methods

In this section, we describe the methods for experimental batch correction that will be evaluated in this paper using the metrics defined in Section 3.2.

4.1. Baseline

The baseline method is a standard convolutional classifier architecture (see Figure 5). A detached batch classification head is added to calculate experimental batch accuracy without backpropagating experimental batch classification error into the rest of the network. For data augmentations, we use horizontal and vertical flips, 90-degree rotations, and cutmix [52]. We train for 100 epochs, using a cosine learning rate schedule with a 5 epoch linear warmup and learning rate of 0.1024, an SGD optimizer with 0.9 momentum, and a batch size of 512 distributed across 8 Nvidia A100 GPUs. Before feeding an image into the network, we preprocess the image with a channel-wise *self-standardization*, *i.e.*, we subtract the mean and divide it by the standard deviation of the image's pixel intensities per channel.

4.2. AdaBN

Adaptive batch normalization (AdaBN) [32] modifies standard batch normalization [25] layers to use statistics from individual domain distributions (e.g., from experimental batch distribution in our case) rather than the entire training set distribution, both during training and at test time. Therefore, during training, it is necessary to sample mini-batches from a single experimental batch at a time. By doing so, the model is able to normalize intermediate features within the context of the experimental batch distribution. The rest of the model is unchanged (see Figure 5).

4.3. Gradient reversal

Gradient reversal [19] is an adversarial method that changes the sign of the gradient for specific layers in the model, e.g., layers connecting the heads of adversarial losses to the rest of the network. Intuitively, this method updates model weights at the gradient reversal layer in order to increase the adversarial loss, while the rest of the head updates its weights in order to decrease the loss, giving rise to the adversarial nature of this method. We want the model to be invariant to differences in experimental batches, thus to implement this method, we reattach the experimental batch classification head mentioned in Section 3.2.3 using gradient reversal (see Figure 5).

4.4. AdaBN + gradient reversal

We also apply adaptive batch normalization and gradient reversal simultaneously in order to evaluate their combined ability to correct experimental batch effects.

5. Experiments

In this section, we evaluate the methods described in Section 4 using the evaluation metrics described in Section 3.2.

5.1. Evaluation metric performance

The results of the experimental batch correction methods are summarized in Table 1. The baseline classifier generalizes poorly to new batches and classifies experimental batches about 30x better than random. The AdaBN model improves experimental batch generalization to nearly 96% while significantly reducing experimental batch classification accuracy (∼8x better than random). Interestingly, gra-

Method	Perturbation classification accuracy (batch-separated)	Perturbation classification accuracy (batch-stratified)	Batch generalization	Batch classification accuracy
Baseline	75.1% ± 0.2%	91.1% ± 0.1%	82.4%	59.2% ± 0.7%
Gradient reversal	71.2% ± 0.4%	89.1% ± 0.1%	79.9%	**1.8%** ± 0.1%
AdaBN	**87.1%** ± 0.2%	**91.1%** ± 0.1%	**95.6%**	16.4% ± 0.3%
AdaBN + gradient reversal	86.2% ± 0.3%	90.2% ± 0.2%	**95.6%**	2.3% ± 0.1%

Table 1. Performance of experimental batch correction methods on the proposed metrics. All models, despite having similar perturbation accuracy on seen batches during training, vary in their ability to generalize to new batches as well as batch classification accuracy. AdaBN significantly improves generalization to new batches, and gradient reversal reduces batch information encoded in embeddings. Using both methods simultaneously yields the benefits of both. For every method, the model was trained 5 times on both batch-separated and batch-stratified splits. For descriptions of the splits, metrics, and methods, see Sections 3.1, 3.2, and 4, respectively.

dient reversal does not improve experimental batch generalization but does reduce experimental batch classification accuracy to random chance. Finally, combining AdaBN and gradient reversal yields the benefits of both methods: top experimental batch generalization, and near-random experimental batch classification.

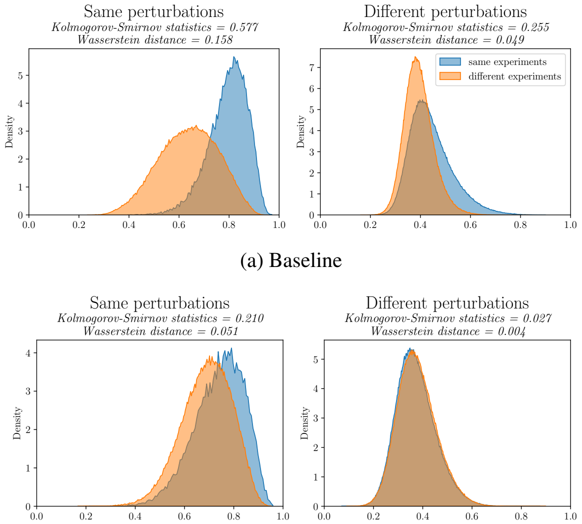
5.2. Visualization of embedding space

In order to gain a better understanding of the information encoded in the embeddings learned by each experimental batch correction method, in Figure 6 we visualize the learned embedding spaces from our baseline, gradient reversal, AdaBN methods using UMAP embeddings [37] (AdaBN + gradient reversal UMAPs are similar to AdaBN UMAPs). We note that while gradient reversal is able to reduce experimental batch classification accuracy to random when trained on the batch-stratified split, this behavior does not generalize well to experimental batches from unseen experiment batches. In contrast, AdaBN is far more effective in aligning unseen experiment batches since it normalizes intermediate image features with the statistics of the associated experimental batch, rather than the statistics of the training set as used in standard batch normalization.

5.3. Preservation of embedding similarities

While the previous section demonstrated that AdaBN is sufficient to align embedding distributions across experimental batches, we also wondered if it would preserve geometric relationships across batches. In order to answer this question, we consider the following distributions of cosine similarities between perturbation embeddings:

1. same perturbations in same experimental batches
2. different perturbations in the same experimental batches
3. same perturbations in different experimental batches (but same cell type)
4. different perturbations in different experimental batches (but same cell type)



(a) Baseline

(b) AdaBN

Figure 7. Distributions of cosine similarities between image embeddings for the **(a)** baseline and **(b)** AdaBN methods. **Blue**: cosine similarities between embeddings from the same experiments. **Orange**: cosine similarities between embeddings from different experiments, but the same cell types. **Left**: cosine similarities between embeddings of the same perturbation. **Right**: cosine similarities between embeddings of different perturbations. Two measures of distributional similarity, the Kolmogorov-Smirnov statistic and Wasserstein distance, are computed between the distributions in each plot. Note that the baseline distributions of the same and different perturbation cosine similarities are distinctly different within and across experimental batches, while the AdaBN distributions are very similar, showing that AdaBN preserves geometric relationships between embeddings even across experimental batches. Note that the cosine similarities are always positive because all values in embeddings are positive as embeddings are obtained by passing features through ReLU in the model.

In Figure 7, we compare distributions 1 and 2 with distributions 3 and 4, for both the baseline and AdaBN methods. The similarity of these pairs of distributions to each other

Model	HUVEC	RPE	HepG2	U2OS
Baseline	84.2 ±0.2	79.0 ±0.4	76.2 ±0.1	26.1 ±1.5
Gradient reversal	83.8 ±0.2	78.1 ±0.5	74.0 ±0.6	24.3 ±0.7
AdaBN	92.1 ±0.2	87.2 ±0.0	86.2 ±0.2	68.2 ±0.1
AdaBN + gradient reversal	92.0 ±0.0	87.5 ±0.1	85.6 ±0.1	66.9 ±0.3

Table 2. Perturbation classification accuracy (%) per cell type. Note that increases in perturbation classification accuracy due to AdaBN are larger for more difficult cell types.

Model	HUVEC	RPE	HepG2	U2OS
Baseline	39.5 ±0.7	39.2 ±2.0	31.4 ±0.4	2.8 ±1.0
Gradient reversal	41.4 ±0.5	38.8 ±0.5	32.3 ±0.4	3.0 ±0.3
AdaBN	55.1 ±1.1	56.1 ±0.4	56.2 ±1.6	44.0 ±0.9
AdaBN + gradient reversal	55.3 ±2.0	56.7 ±0.9	55.5 ±0.6	44.1 ±1.4

Table 3. Perturbation classification accuracy (%) per cell type on simplified training sets containing only 3 experiments of a single cell type. HUVEC, RPE, and HepG2 cell types are easier to learn than U2OS, however, AdaBN significantly improves all classification accuracies, especially U2OS.

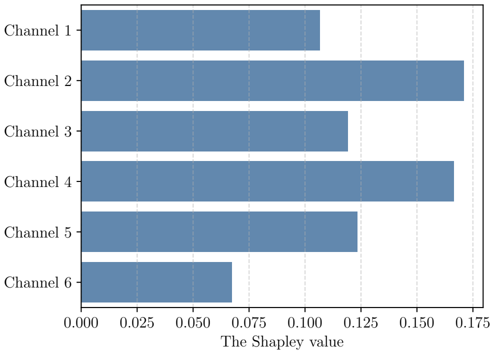


Figure 8. The Shapley values for each channel using the baseline method, represent contributions to perturbation classification accuracy. Higher values represent greater importance. Similar results are observed for each experimental batch correction method.

Normalization	Accuracy
All images	60.8 ± 1.1
Control images per experiment	68.4 ± 0.5
All images per experiment	68.6 ± 0.5
Control images per plate	73.4 ± 0.3
All images per plate	73.4 ± 0.5
Self-standardization	$\mathbf{75.1} \pm 0.2$

Table 4. Perturbation classification accuracy (%) for different image normalization methods using the baseline method. Self-standardization, where each channel of a single image is standardized by its own mean and standard deviation, yields the best results.

would be strong evidence that the experimental batch correction method preserves geometric relationships across experimental batches. To this end, we calculate two measures of distributional similarity, the Kolmogorov-Smirnov (KS) statistic and Wasserstein Distance (WD), between each pair of distributions in order to quantify these similarities. As can be seen, the baseline distributions of the same and different perturbation cosine similarities are distinctly different within and across experimental batches, indicating that geometric relationships are not preserved across experimental batches for the baseline method. In contrast, the AdaBN distributions are very similar within and across experiment batches, demonstrating that AdaBN does indeed preserve geometric relationships across experimental batches. In Supplementary Table S3, we calculate these similarity metrics for all batch correction methods.

5.4. Classification accuracy per cell type

Table 2 shows perturbation classification accuracy for each of the four cell types. Note that HUVEC accuracies are highest, followed by RPE and HepG2, and finally U2OS. This is in line with the differing proportions of experimental batches per each cell type in the training set. In order to obtain a more fair comparison of per-cell perturbation classification accuracy, we randomly selected 3 experimental batches for each cell type from the original training set to form a new training set. The results are shown in Table 3. We note that the HUVEC, RPE, and HepG2 cell types are far easier to learn than U2OS; however, AdaBN significantly improves classification accuracies in all cell types, especially U2OS. Comparing the results (for U2OS since training sets were the same in both) in Tables 3 and 2, we conclude that jointly training a method on all cell types rather than individual cell types greatly improves perturbation classification accuracy. We therefore reason that the poorer performance on U2OS is not due entirely to the lower number of U2OS experimental batches in the dataset, but possibly because U2OS is biologically distinct from the other cell types, and expresses different, less consistent phenotypes than the others do.

5.5. Channel importance

In Figure 8, we study the importance of each channel by plotting its Shapley value [46] for the baseline method. Shapley values measure the relative contributions each channel makes when assigning correct classes to our (batch-separated) test set. Note that Channels 2 and 4 are the most important, while Channel 6 is the least important.

In fact, using only the first five channels improves perturbation accuracy over using all channels by 2% in the baseline method (see Supplementary Material for this and other channel subset accuracy results).

5.6. Image preprocessing

We tried different image normalization methods for preprocessing the images. In all cases, we calculate per-channel means and standard deviations on different subsets of the dataset and standardize (*i.e*., subtract the mean and divide by the standard deviation) each image with those statistics before using them as input to the networks. The (batch-separated) perturbation classification accuracies associated with each normalization method are presented in Table 4. Self-standardization (standardization using only the image itself) outperforms other methods by a significant margin. Interestingly, the self-standardization improves perturbation classification accuracy by more than 14 percentage points compared to the standard computer vision practice of using global statistics calculated from the entire training set. We hypothesize that this margin is due to the uniform background of RxRx1 images across experimental batches, which contains little biological information but whose size relative to the foreground of cells can change dramatically from perturbation to perturbation and even image to image. Thus image-level statistics are proportional to the cellular content of an image, so that self-standardization normalizes each image to the common scale of the average cell contained within the image.

6. Conclusion and future directions

In this paper, we described the *RxRx1* dataset, an image dataset systematically designed to study experimental batch effect correction methods. The dataset contains 125,510 6-channel, high-resolution fluorescence microscopy images of human cells under 1,138 genetic perturbations in 51 experimental batches across 4 cell types. We proposed a task and several metrics to evaluate the performance of different experimental batch correction methods. We demonstrated that while both adaptive batch normalization (AdaBN) [32] and gradient reversal [19] are effective techniques for removing experimental batch information from image embeddings, only AdaBN was effective in generalizing to unseen experimental batches, due to the manner in which it normalizes all intermediate feature maps using statistics from the corresponding experimental batch. We also calculated the importance of each image channel in this task, and the value of self-standardization as an image preprocessing step. We hope that the introduction of the RxRx1 dataset will encourage further research into the complex problem of correcting experimental batch effect, as well as other issues that arise in the analysis of high-throughput screening data.

6.1. Future directions

There are several methodologies for extracting features from microscopy imaging screens, including traditional feature extraction (*i.e*. CellProfiler) [33, 47], leveraging pre-trained deep learning models [1, 42], and training deep learning models on microscopy images directly [20]. As both AdaBN and gradient reversal are deep learning methodologies, it is not possible to directly apply these methods to traditional feature extraction pipelines, yet an appropriate comparison would be useful to understand the benefit of end-to-end feature training. In the future, we plan to provide such a comparison, e.g., train neural networks on CellProfiler features with AdaBN and gradient reversal.

Our approach relies on weakly supervised learning [9, 38] since we train models to predict the experimental perturbation in each well, without validating that each treatment induces a unique visual phenotype (N.B.: such validation is likely impossible). This means that there might be multiple perturbations that either do not perturb the cellular morphology or perturb it in similar ways to other perturbations, yet the perturbation classification task would reward distinguishing them. This would encourage reliance on spurious features or correlation, which inhibits learning image representations that capture meaningful morphological features. Recently, self-supervised methods have been shown to match the performance of supervised models on natural image computer vision tasks [3, 10, 12]. Applying such training techniques for microscopy screening data [13, 43] represents a potentially fruitful future direction for this work.

Finally, we acknowledge that the proposed perturbation classification task groups any morphological variation not associated with a common perturbation under the umbrella term *experimental batch effect*, which is usually reserved for technical effects only. One could imagine improving the task in a way that would not penalize intrinsic morphological features, like those associated with cell type differences, even if they are not associated with variations amongst perturbations. Such a task would promote the development of more effective experimental batch correction methods that better disentangle biological and technical causal factors, and we hope to provide such an update to this work in the future.

7. Potential societal impact

We do not foresee any negative impacts arising from the specific contributions of this paper. The data was produced using commercially-available primary and immortalized cell lines. The RxRx1 dataset is intended for the development of methods that further our understanding of biology, genetics, and pharmacology, and their application in drug discovery.

References

[1] D Michael Ando, Cory Y McLean, and Marc Berndl. Improving phenotypic measurements in high-content imaging screens. *BioRxiv*, page 161422, 2017. 2, 8

[2] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular systems biology*, 12(7):878, 2016. 2

[3] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. *arXiv preprint arXiv:2204.07141*, 2022. 8

[4] VARIATION ACROSS MANY EXPERIMENTAL BATCHES. Rxrx1: An image set for cellular morphological. 2

[5] Michael Boutros, Florian Heigwer, and Christina Laufer. Microscopy-based high-content screening. *Cell*, 163(6):1314–1325, 2015. 1

[6] Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9):1757–1774, 2016. 3

[7] James R Broach and Jeremy Thorner. High-throughput screening for drug discovery. *Nature*, 384(6604 Suppl):14–16, 1996. 1

[8] Juan C Caicedo, Sam Cooper, Florian Heigwer, Scott Warchal, Peng Qiu, Csaba Molnar, Aliaksei S Vasilevich, Joseph D Barry, Harmanjit Singh Bansal, Oren Kraus, et al. Data-analysis strategies for image-based cell profiling. *Nature methods*, 14(9):849–863, 2017. 2

[9] Juan C Caicedo, Claire McQuin, Allen Goodman, Shantanu Singh, and Anne E Carpenter. Weakly supervised learning of single-cell feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9309–9318, 2018. 8

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 8

[11] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018. 2

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 8

[13] Jan Oscar Cross-Zamirski, Guy Williams, Elizabeth Mouchet, Carola-Bibiane Schönlieb, Riku Turkki, and Yinhai Wang. Self-supervised learning of phenotypic representations from cell images with weak labels. *arXiv preprint arXiv:2209.07819*, 2022. 8

[14] Jaeger Davis, Steve P Crampton, and Christopher CW Hughes. Isolation of human umbilical vein endothelial cells (huvec). *JoVE (Journal of Visualized Experiments)*, (3):e183, 2007. 2

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2

[16] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. Innovation in the pharmaceutical industry: new estimates of r&d costs. *Journal of health economics*, 47:20–33, 2016. 1

[17] María Teresa Donato, Laia Tolosa, and María José Gómez-Lechón. Culture and functional characterization of human hepatoma hepg2 cells. In *Protocols in In Vitro Hepatocyte Research*, pages 77–93. Springer, 2015. 3

[18] Christophe J Echeverri and Norbert Perrimon. High-throughput rnai screening in cultured cells: a user's guide. *Nature Reviews Genetics*, 7(5):373–384, 2006. 1

[19] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 5, 8

[20] William J Godinez, Imtiaz Hossain, Stanley E Lazic, John W Davies, and Xian Zhang. A multi-scale convolutional neural network for phenotyping high-content cellular images. *Bioinformatics*, 33(13):2010–2019, 2017. 8

[21] Wilson Wen Bin Goh, Wei Wang, and Limsoon Wong. Why batch effects matter in omics data, and how to avoid them. *Trends in biotechnology*, 35(6):498–507, 2017. 2

[22] Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018. 2

[23] Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature biotechnology*, 37(6):685–691, 2019. 2

[24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5

[26] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 2

[27] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019. 2

[28] Oren Z Kraus, Jimmy Lei Ba, and Brendan J Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016. 2

[29] Oren Z Kraus, Ben T Grys, Jimmy Ba, Yolanda Chong, Brendan J Frey, Charles Boone, and Brenda J Andrews. Automated analysis of high-content microscopy data with deep learning. *Molecular systems biology*, 13(4):924, 2017. 2

[30] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010. 2

[31] Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature communications*, 11(1):1–14, 2020. 2

[32] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016. 5, 8

[33] Vebjorn Ljosa, Peter D Caie, Rob Ter Horst, Katherine L Sokolnicki, Emma L Jenkins, Sandeep Daya, Mark E Roberts, Thouis R Jones, Shantanu Singh, Auguste Genovesio, et al. Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. *Journal of biomolecular screening*, 18(10):1321–1329, 2013. 8

[34] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018. 2

[35] Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J Theis, and F Alexander Wolf. Conditional out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics*, 36(Supplement 2):610–617, 12 2020. 2

[36] Ricardo Macarron, Martyn N Banks, Dejan Bojanic, David J Burns, Dragan A Cirovic, Tina Garyantes, Darren VS Green, Robert P Hertzberg, William P Janzen, Jeff W Paslay, et al. Impact of high-throughput screening in biomedical research. *Nature reviews Drug discovery*, 10(3):188–195, 2011. 1

[37] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 6

[38] Nikita Moshkov, Michael Bornholdt, Santiago Benoit, Claire McQuin, Matthew Smith, Allen Goodman, Rebecca Senft, Yu Han, Mehrtash Babadi, Peter Horvath, et al. Learning representations for image-based profiling of perturbations. *bioRxiv*, 2022. 8

[39] Katerina N Niforou, Athanasios K Anagnostopoulos, Konstantinos Vougas, Christos Kittas, Vassilis G Gorgoulis, and George T Tsangaris. The proteome profile of the human osteosarcoma u2os cell line. *Cancer genomics & proteomics*, 5(1):63–77, 2008. 3

[40] Vegard Nygaard, Einar Andreas Rødland, and Eivind Hovig. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*, 17(1):29–39, 2016. 2

[41] Hilary S Parker and Jeffrey T Leek. The practical effect of batch on genomic prediction. *Statistical applications in genetics and molecular biology*, 11(3), 2012. 2

[42] Nick Pawlowski, Juan C Caicedo, Shantanu Singh, Anne E Carpenter, and Amos Storkey. Automating morphological profiling with generic deep convolutional networks. *BioRxiv*, page 085118, 2016. 8

[43] Alexis Perakis, Ali Gorji, Samriddhi Jain, Krishna Chaitanya, Simone Rizza, and Ender Konukoglu. Contrastive learning of single-cell phenotypic representations for treatment classification. In *International Workshop on Machine Learning in Medical Imaging*, pages 565–575. Springer, 2021. 8

[44] Jack W Scannell, Alex Blanckley, Helen Boldon, and Brian Warrington. Diagnosing the decline in pharmaceutical r&d efficiency. *Nature reviews Drug discovery*, 11(3):191–200, 2012. 1

[45] Uri Shaham, Kelly P Stanton, Jun Zhao, Huamin Li, Khadir Raddassi, Ruth Montgomery, and Yuval Kluger. Removal of batch effects using distribution-matching residual networks. *Bioinformatics*, 33(16):2539–2546, 2017. 2

[46] Lloyd S Shapley. Notes on the n-person game—ii: The value of an n-person game.(1951). *Lloyd S Shapley*, 1951. 7

[47] Shantanu Singh, M-A Bray, TR Jones, and AE Carpenter. Pipeline for illumination correction of images for high-throughput microscopy. *Journal of microscopy*, 256(3):231–236, 2014. 8

[48] Charlotte Soneson, Sarah Gerster, and Mauro Delorenzi. Batch effect confounding leads to strong bias in performance estimates obtained by cross-validation. *PloS one*, 9(6):e100335, 2014. 2

[49] David C Swinney and Jason Anthony. How were new medicines discovered? *Nature reviews Drug discovery*, 10(7):507–519, 2011. 1

[50] Thomas Tuschl. Rna interference and small interfering rnas. *Chembiochem*, 2(4):239–245, 2001. 3

[51] Song Yang, Jun Zhou, and Dengwen Li. Functions and diseases of the retinal pigment epithelium. *Frontiers in Pharmacology*, page 1976, 2021. 2

[52] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 5

[53] Yuexin Zhou, Shiyou Zhu, Changzu Cai, Pengfei Yuan, Chunmei Li, Yanyi Huang, and Wensheng Wei. High-throughput screening of a crispr/cas9 library for functional genomics in human cells. *Nature*, 509(7501):487–491, 2014. 1