Self-supervision advances morphological profiling by unlocking powerful image representations

Vladislav Kim[1*], Nikolaos Adaloglou[1,2*], Marc Osterland[1], Flavio M. Morelli[1], Paula A. Marin Zapata[1§]

[1]Machine Learning Research, Bayer AG, Berlin, Germany

[2]Mathematical Modeling of Biological Systems, Heinrich Heine University Düsseldorf, Düsseldorf, Germany

[*]Equal contributions

[§]Corresponding author

Abstract

Morphological profiling is a powerful technology that enables unbiased characterization of cellular states through image-based screening. Inspired by recent progress in self-supervised learning (SSL), we sought to explore the potential benefits of using SSL in this domain and conducted a comprehensive benchmark study of recent SSL methods for learning representations from Cell Painting images without segmentation. We trained DINO, MAE, and SimCLR on subsets of the JUMP-CP consortium data, one of the largest publicly available Cell Painting image sets, and observed improved model performance with larger and more heterogeneous training sets. Our best model (DINO) surpassed the widely used profiling tool CellProfiler by 29% in mean average precision (mAP) on classifying chemical perturbations and significantly accelerated feature extraction by 50x, at a lower cost. Moreover, DINO outperformed CellProfiler in clustering gene families on an independent gene overexpression dataset. Our findings indicate that SSL methods can improve the efficiency and performance of morphological profiling, offering the potential to expedite drug discovery and reduce compute costs.

Introduction

Morphological profiling uses image-based readouts to characterize the effect of chemical and genetic perturbations[1–4] based on cell morphology. Offering high throughput and low cost, this technology has numerous applications in drug discovery such as mode of action identification[5,6], off-target effect detection[7,8], drug repurposing[9–11] and toxicity prediction[12]. One widely used assay for morphological profiling is Cell Painting, which uses 5 fluorescent markers to stain 8 cellular compartments[13]. Despite significant progress in the visual representation learning field[14–17], the analysis of Cell Painting images still largely relies on classical computer vision approaches[18].

The classical morphological profiling workflow starts with single-cell segmentation, which is typically carried out using threshold-based methods as implemented in CellProfiler[18] or similar software tools[19,20]. The segmented cells are characterized using hand-crafted descriptors such as shape, size, intensity and texture[21] among others. The descriptors are then aggregated to obtain a single vector of morphological features for each probed condition. Since morphological features are highly intercorrelated, feature selection methods are usually applied to reduce redundancy[21]. This multi-step workflow is computationally intensive as it often requires adjustment of segmentation and feature selection parameters when applied to new datasets. By contrast, deep learning models can offer a computationally efficient and segmentation-free

alternative for morphological profiling. These models leverage data-driven descriptors to potentially enhance the generalization and sensitivity of learned representations beyond what can be achieved with hand-crafted features.

Several deep-learning-based approaches have been proposed for feature extraction from microscopy images. Some of them use transfer learning[22], image inpainting[23], variational autoencoders[24], supervised[25], and weakly supervised learning[26–28] for single-cell feature extraction. However, these methods require segmentation to locate single cells. Other approaches extract representations from whole images using supervised learning[29,30], transfer learning[31] or a combination of self- and weakly-supervised learning[32]. These approaches are limited by the availability of training labels or the dependence on pretrained weights (e.g. ImageNet-pretrained weights), which restricts their application to 3-channel images or requires embedding concatenation for multichannel (>3) applications. To date, only few approaches [33,34] learn directly from microscopy images without the need for segmentation or manually curated annotations.

The JUMP-CP consortium recently released one of the largest public Cell Painting datasets[35]. This large-scale image set (200 TB) with approximately 116,000 compounds and 13,000 genetic perturbations (CRISPR and ORF) offers unique opportunities for developing novel AI-based morphological profiling methods. Most of the compounds probed by the JUMP-CP consortium lack annotations, with only 4.5% of the screened compounds having experimentally elucidated biological activity. Thus, leveraging the full extent of this large image corpus demands methods that do not rely on data curation or biological annotations.

Self-supervised learning (SSL) is a class of powerful methods that aim to learn feature representations from unlabeled data through a pretext task. Early pretext tasks focused on predicting image transformations such as rotations[36] or jigsaw puzzles[37]. However, the current state-of-the-art performance has been achieved through joint-embedding methods that maximize the agreement between transformed views of the same image. For instance, the joint-embedding methods PIRL[38], MoCo[39] and SimCLR[15] use a contrastive loss to match paired views ("positives") from the same image and repel unrelated views ("negatives") from different images in the representation space. However, for optimal performance, contrastive methods require large minibatch sizes or memory banks, which can be computationally demanding. This limitation has been overcome by recent non-contrastive approaches such as BYOL[40] and DINO[16]. These approaches train a student network to predict the output of a teacher network while receiving different augmented views of an input image. Remarkably, DINO has been one of the best-performing SSL approaches across different domains[16,41,42].

Recent advancements in the SSL field have been accelerated by the adoption of the vision transformer[43] (ViT) architecture. Vision transformers operate on projected image patches and use the self-attention mechanism[44] to capture global and local relationships between image patches. The high computational cost of training ViT architectures inspired novel reconstruction-based pretext tasks, such as image masking, which provides a strong supervisory signal and improves training efficiency. This has been demonstrated by masked autoencoders[17] (MAE) and masked Siamese networks[45] (MSN) which achieved state-of-the-art results in the natural image domain. Notably, ViT performance scales favorably with data volume and complexity[46], which makes these models particularly suitable for high-throughput imaging data.

We present, for the first time, a comprehensive benchmark study of state-of-the-art SSL methods adapted for Cell Painting images. We trained all SSL models directly in the applicability domain, unlike previous studies which relied on pretrained ImageNet weights. Using the JUMP-CP data, we assessed all methods on evaluation sets with both compound and genetic perturbations. Most importantly, we explored the requirements for the successful application of SSL in the fluorescence microcopy domain including relevant image augmentations, feature postprocessing techniques, and scalability of SSL models with data and architecture size/complexity. Our results indicate that the DINO framework outperforms other models and can serve as a segmentation-free alternative to CellProfiler, with significantly faster processing speed.

Results

SSL framework for segmentation-free morphological profiling

We first sought to establish a framework for training and evaluating self-supervised learning (SSL) models on microscopy images. To this end, we adapted 3 state-of-the-art SSL approaches (SimCLR[15], DINO[16] and MAE[17]) to extract feature representations from 5-channel Cell Painting images (Fig. 1a). The representations learned via self-supervision were used to construct morphological profiles of chemical and genetic perturbations and were benchmarked against two baselines (Fig. 1b): CellProfiler[18], a widely used computational tool for morphological profiling, and transfer learning from a model pretrained on natural images using a supervised classification task[43].

Model training

Our self-supervised learning framework operates directly on image crops and does not require the segmentation of individual cells. During training, we randomly sample a 224x224 image crop from each microscopy image. To ensure that each training input contains cells, we select only those image crops with at least 1% of the non-empty area in the nucleus-staining channel (see Methods). For the joint-embedding methods SimCLR and DINO, multiple image crops (views) are sampled from each microscopy image and transformed using a stochastic augmentation pipeline (see Methods and "Establishing augmentations for Cell Painting images"). We used the small (ViT-S) and base (ViT-B) vision transformer architectures and trained all SSL models starting from random initializations. In addition to the SSL loss, we tracked mean replicate correlation (Supplementary Fig. 1) to monitor model performance over training epochs and selected the checkpoint with the highest mean replicate correlation for each model (see Methods).

Model inference

A representation learning model maps an input image *x* to a *d*-dimensional embedding space $\{f: x \rightarrow R^d\}$. In joint-embedding methods (SimCLR and DINO), the mapping $f(\cdot)$ is performed by the backbone network, while in a masked autoencoder (MAE) the encoder network acts as a mapping function. The dimensionality of learned representations depends on the network architecture, with $d = 384$ for ViT-S and $d = 768$ for ViT-B.

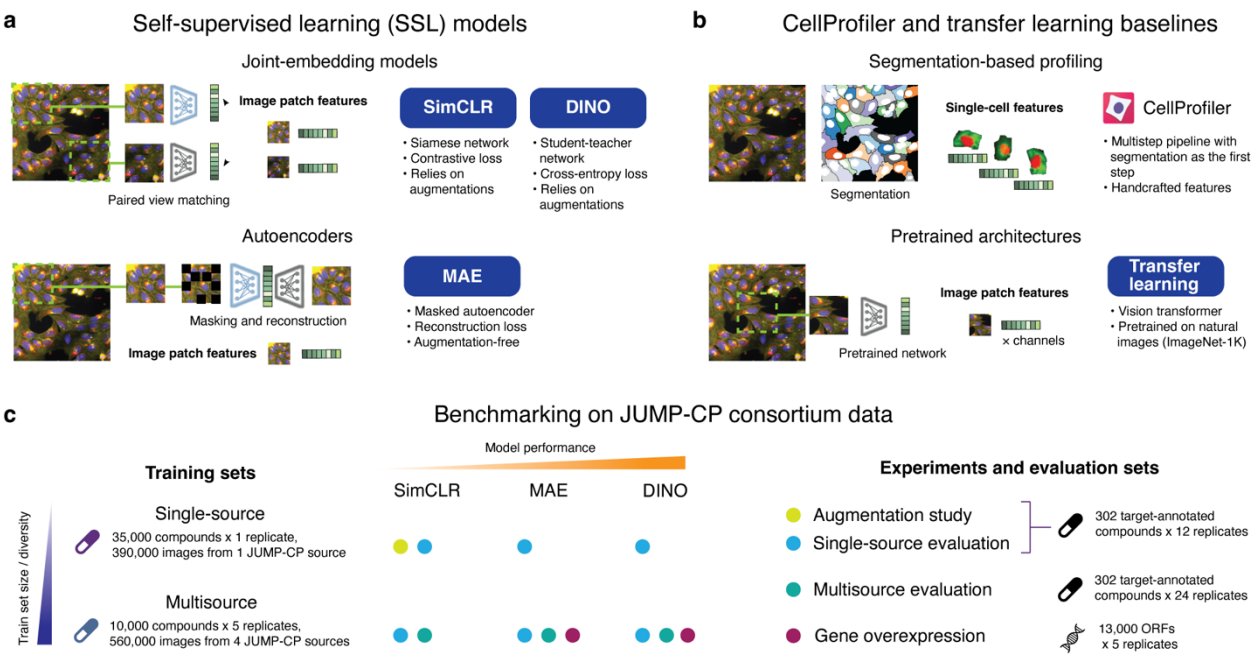

Figure 1: Self-supervised learning for morphological profiling.
a) Schematic of the three SSL models used in this study. All models are segmentation-free and use solely image crops as input. SimCLR and DINO were trained on the pretext task of matching representations from augmented views of the same image. MAE was trained on the image reconstruction task with partially masked input. **b)** Methods considered as baselines. CellProfiler: conventional method based on single-cell segmentation and handcrafted features. Transfer learning: pretrained vision transformer that outputs per-channel representations. **c)** Overview of the JUMP-CP consortium data subsets used for training and evaluation of SSL models. SSL models were trained on two training sets: a single-source and a multisource set. The colored points indicate which evaluation sets were used to assess the models trained on the single-source and multisource data. The image augmentation study was conducted only for SimCLR trained and evaluated on the single-source data.

At inference time, each microscopy image I, corresponding to a single field of view (FOV), was split into 224x224 image crops $\{x_i, i = 1, \ldots, N_{crops}\}$. All image crops were passed through the backbone or encoder to generate crop embeddings $f(x_i)$ that were aggregated using the arithmetic mean to produce an image embedding $f(I)$. Crops with no cells were excluded following the same criteria as during training. Finally, we obtain a well embedding by computing the mean feature vector across all fields of view imaged for each well.

Postprocessing of learned representations

Microscopy images suffer from technical biases which reduce the comparability of feature representations. To correct for these effects, we tested several postprocessing methods (see Methods) including typical variance normalization[22] (sphering), z-scoring (standardize), and robust z-scoring (MAD robustize). The choice of optimal normalization procedure was crucial for all representations (Supplementary Fig. 2). The best postprocessing strategy for all SSL representations consisted of “sphering” followed by “MAD robustize”, while the best strategy for CellProfiler and transfer learning was “MAD robustize” alone. We normalize well-level features using their respective best strategy for all remaining experiments.

Following the common practices in the field, we applied a variance threshold to remove the low-variance features for all representations (see Methods). Moreover, to select an optimal feature set for each approach, we investigated the effect of adding a correlation threshold to remove redundant features. For CellProfiler, we additionally included the feature set reported in [47] which was obtained by applying variance and correlation thresholds on a highly diverse dataset (CPJUMP1 dataset) with chemical and genetic perturbations. Our results (Supplementary Fig. 3) demonstrate that the correlation threshold did not influence the performance of SSL representations as it only removed 1 feature from MAE and DINO representations. However, it improved the performance of CellProfiler and reduced the dimensionality 5-fold. Nevertheless, the best CellProfiler features corresponded to the CPJUMP1 set. Based on these findings, we chose the CPJUMP1 feature set for CellProfiler and variance-threshold features for SSL representations in all our experiments.

Finally, after feature selection and normalization, we aggregated well profiles by taking the mean across multiple technical replicates of the same perturbation. We used several aggregation levels: *batch-aggregated profiles* average all replicates within a batch, *source-aggregated profiles* average all replicates from all baches within a source, and *consensus profiles* average all replicates within a dataset.

Evaluating learned representations

We evaluated image representations based on two key criteria: reproducibility (agreement between replicates of the same perturbation) and biological relevance (agreement between perturbations with the same biological annotation). We quantify both criteria with binary k-nearest neighbors' classifiers for each sample (match vs no match) using perturbation and target information as labels.

To this end, we compute two types of metrics for each criterion. The first, *mean average precision* (*mAP)*, summarizes the precision-recall curve over a range of *k* nearest neighbors (see Methods), independently for perturbation and target labels. *Perturbation mAP* values were estimated on batch-aggregated profiles, thus quantifying the ability to match perturbations across batches. *Target mAP* values were estimated on consensus profiles, focusing on biological content after technical variations were averaged out. For *mAP* calculation, average precision (*AP*) was averaged across all samples. We also tested alternative *AP* aggregation strategies (Supplementary Fig. 4), which produced similar results.

The second class of metrics reports the nearest neighbor (NN) accuracy estimated on well profiles with restrictions on the possible match. The *not-same-batch (NSB) accuracy* restricts true positive matches to profiles from different batches, while *not-same-batch-or-perturbation (NSBP) accuracy* restricts true positive matches to profiles from different batches and distinct perturbations. We used *perturbation NSB accuracy* to evaluate the reproducibility of well profiles across batches, and *target NSBP accuracy* to evaluate biological relevance.

JUMP-CP datasets for benchmarking SSL representations

Each compound in the JUMP-CP dataset[35] was screened in 5 replicates across 12 laboratories ("sources") using a standardized protocol but possibly different instrumentation, which allowed us to study the scaling properties of SSL models in relation to training data size and

heterogeneity. We used two training sets (Fig. 1c): a single-source training set comprised of images solely from source 3 and a multisource training set with images from sources 2, 3, 6, and 8 (see Methods). During model training, we used only images of cells treated with chemical perturbations.

We evaluated all methods on 3 subsets of the JUMP-CP data: 1) a single-source evaluation set 2) a multi-source evaluation set and 3) a gene overexpression set (Fig. 1c). The single-source and multisource evaluation sets were constructed using JUMP Target2 plates (see Methods and [35]) from the respective sources of the single-source and multisource training sets. JUMP Target2 plates contain compounds with target annotations and were imaged in every experimental batch, which enabled us to not only assess model performance using biological labels but also evaluate batch and laboratory effects. Finally, the gene overexpression set was used as a test set to assess the performance of the best SSL models determined in the single-source and multisource evaluations. This set corresponds to a selection of active Open Reading Frames (ORF) imaged at source 4 (see Methods). The inclusion of genetic perturbations from a new source allowed us to test the models' ability to generalize to previously unseen perturbations.

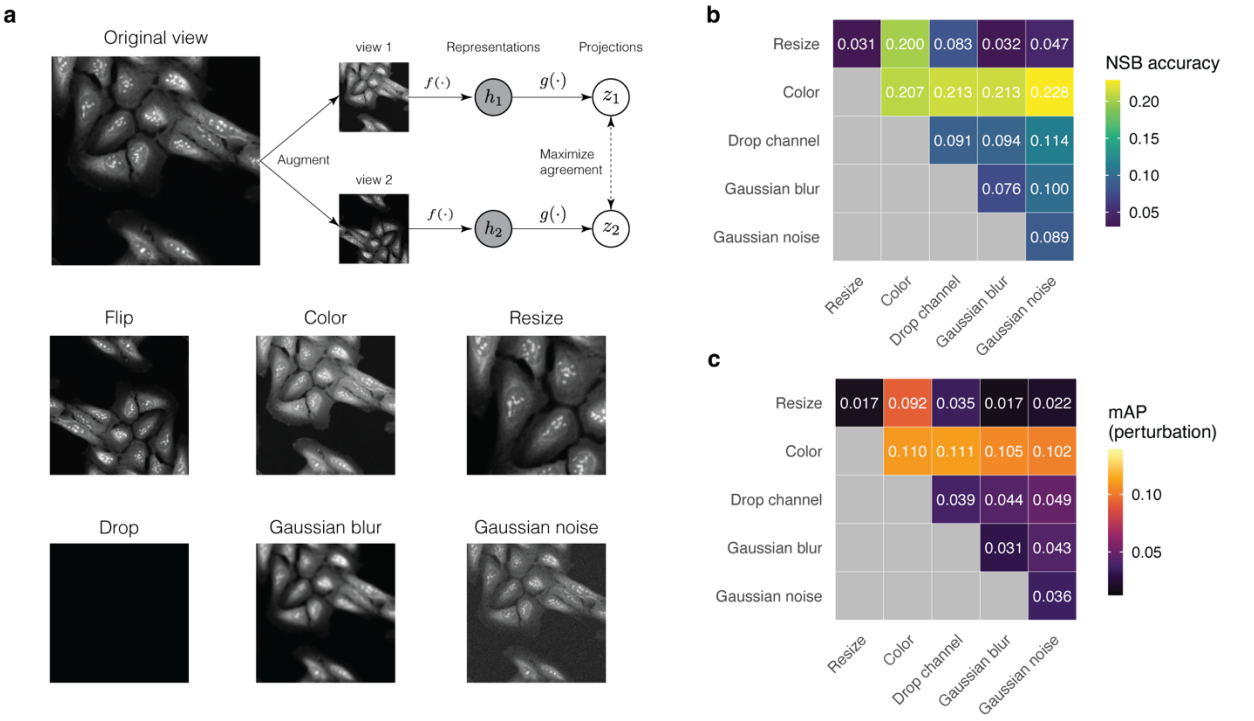

Figure 2: Augmentations for multichannel microscopy images.
a) Schematic of the SimCLR contrastive learning framework and representative images of the tested augmentations. Flip: vertical and horizontal flips. Color: per-channel stochastic intensity shift and brightness adjustment. Resize: random crop of variable dimensions followed by rescaling to a fixed crop size. Drop: omit one color channel at random. Gaussian blur: Gaussian kernel smoothing. Gaussian noise: addition of Gaussian noise. For details, refer to the Methods section. **b)-c)** Performance comparison of SimCLR models with 15 pairwise combinations of augmentations trained and evaluated on the single-source datasets. Diagonal entries report the performance of individual augmentations. The 'Flip' augmentation was applied by default. Performance is assessed based on the reproducibility metrics *not-same-batch accuracy* (*NSB*) and *perturbation mean average precision* (*mAP*).

Establishing augmentations for Cell Painting images

Joint-embedding methods such as SimCLR and DINO rely on image augmentations. While the relative importance of individual augmentations was previously determined for RGB natural images[15], there are no prior works systematically assessing augmentations for multichannel microscopy images. Therefore, we conducted a comprehensive study for Cell Painting images to evaluate the contribution of several common augmentations. We followed a similar study design as in [15], probing all pairwise combinations of 5 augmentations (Fig. 2a): 'Resize', 'Color', 'Drop channel', 'Gaussian noise' and 'Gaussian blur' (see Methods). Since color jittering[15,16] is specific to RGB images, we replaced the 'Color' augmentation with a more general transform consisting of random brightness change and intensity shift applied on each fluorescent channel independently (see Methods). Moreover, a 'Flip' augmentation was applied by default. For comparability and computational efficiency, we used the same architecture as in [15] (SimCLR with a ResNet-50 backbone) and train and evaluate all models on the smaller, single-source datasets.

A pairwise comparison of augmentation transforms (Fig. 2b-c) showed that the 'Color' augmentation had the greatest positive impact on performance both in terms of *perturbation mAP* and *NSB accuracy*. The 'Resize' operation, which generates image crops at different scales (see Methods), led to a decrease in performance compared to other augmentations. The remaining 3 augmentations ('Drop channel', 'Gaussian noise', 'Gaussian blur') had a negligible effect on *NSB accuracy* and *mAP* relative to the 'Color' augmentation alone. Based on these findings, we established an augmentation pipeline for Cell Painting images that relies primarily on 'Color' and 'Flip' augmentations and excluded all other transforms which didn't improve performance. This pipeline was adopted for DINO and SimCLR models in all subsequent experiments.

DINO trained on multisource data outperforms CellProfiler

We conducted a benchmark study to compare three SSL approaches (SimCLR, DINO, and MAE) and two baseline methods (CellProfiler and transfer learning) using our established workflow and augmentation pipeline. To assess the impact of training set size and model architecture on the performance of these methods, we trained and evaluated all models using our single-source and multisource datasets (see "JUMP-CP datasets for benchmarking SSL representations") and used the small (ViT-S) and base (ViT-B) vision transformer architectures.

We evaluated feature representations of all approaches and found that DINO achieved the best results among the SSL methods, surpassing CellProfiler when trained on sufficient data. Our analysis indicated that DINO representations displayed better reproducibility (Fig. 3a,c) and biological relevance (Fig. 3b,d) compared to their MAE or SimCLR counterparts, with SimCLR producing the worst representations among the SSL approaches. When trained on multisource data, DINO outperformed all other models, surpassing CellProfiler by a margin of 16/29% in *perturbation mAP*, 3/6% in *NSB accuracy,* 22/61% in *target mAP,* and 32/11% in *NSBP accuracy* on the single-source/multisource evaluation sets (Figure 2a-d). This performance gap was even more evident from the F1-score curves over a range of *k* nearest neighbors (Supplementary Fig. 5). The superiority of DINO representations was further confirmed by quantification of reproducibility across JUMP-CP sources (Supplementary Fig. 6). DINO

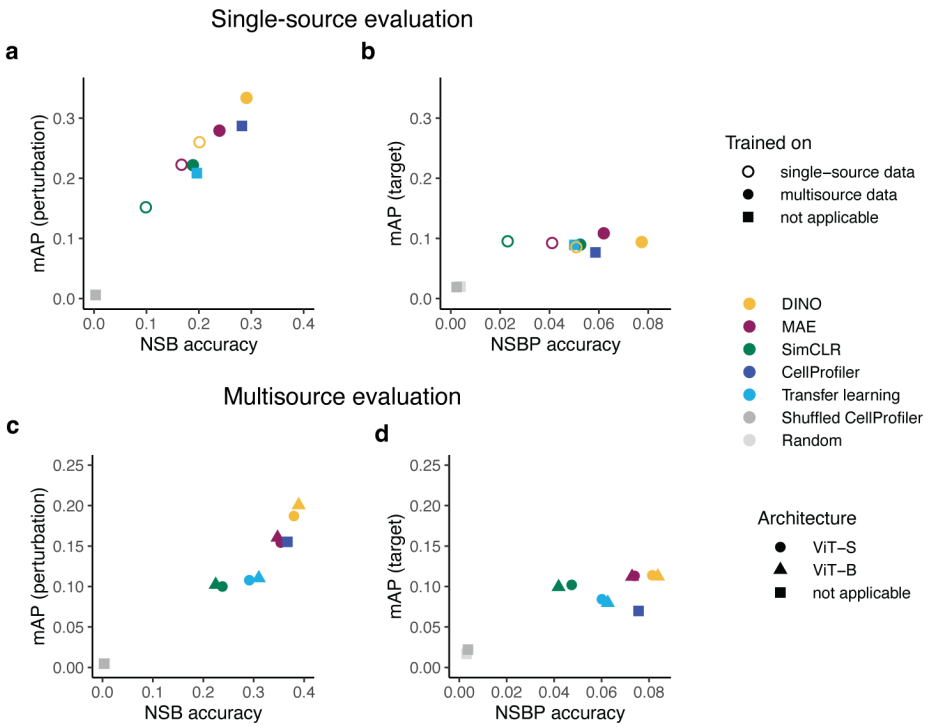

Figure 3: Impact of dataset heterogeneity and architecture size on SSL model performance. Performance comparison of SSL models (DINO, MAE, and SimCLR) and two baselines (CellProfiler and transfer learning) for different datasets (single-source and multisource) and model architectures (ViT-S and ViT-B). Left panels: reproducibility metrics *perturbation not-same-batch (NSB) accuracy* and *mean average precision (mAP).* Right panels: biological relevance metrics *target not-same-batch-or-perturbation (NSBP) accuracy* and *mean average precision (mAP)*. Colors indicate different models and two randomized baselines: Shuffled CellProfiler (CellProfiler features with shuffled labels) and Random (random normally distributed features). **a)-b)** Performance on the single-source evaluation set. Shapes indicate different training sets. Only ViT-S architectures are reported. **c)-d)** Performance on the multisource evaluation set. All SSL models were trained on the multisource training set. Shapes indicate the ViT architecture.

representations yielded similar or better performance to CellProfiler for 3 out of 4 sources (Supplementary Fig. 6a-d) and achieved higher *perturbation mAP* (+73%) and *NSB accuracy* (+2%) on a cross-source matching task based on source-aggregated profiles (Supplementary Fig. 6e). Notably, transfer learning had the worst performance overall, encouraging the use of self-supervised learning for morphological profiling.

The comparison of SSL models trained on the single- and multisource training sets revealed the importance of training set size and heterogeneity for learning meaningful representations. While all SSL models trained on single-source data performed worse than CellProfiler (Fig. 3a,b), some models match (MAE) or exceed (DINO) CellProfiler performance when trained on the multisource dataset. Surprisingly, models based on the larger ViT-B architecture only showed a marginal improvement over the ViT-S architecture (Fig. 3c,d). This shows that the biggest performance gain was achieved by scaling the training set to a more extensive and diverse image set as opposed to increasing the model size. We, therefore, utilized ViT-S architectures in the remaining experiments motivated by their lower memory usage and computational costs.

UMAP projections reveal biological and technical axes of variation

Next, we compared SSL models qualitatively using UMAP dimensionality reduction[48] (see Methods) to visualize DINO, MAE, SimCLR and CellProfiler representations. Based on the results of our benchmark study, we only considered models with ViT-S architectures trained on the multisource dataset. To capture the effects of technical variation such as batch and source (laboratory) effects, all UMAP projections were computed on well profiles of the multisource evaluation set.

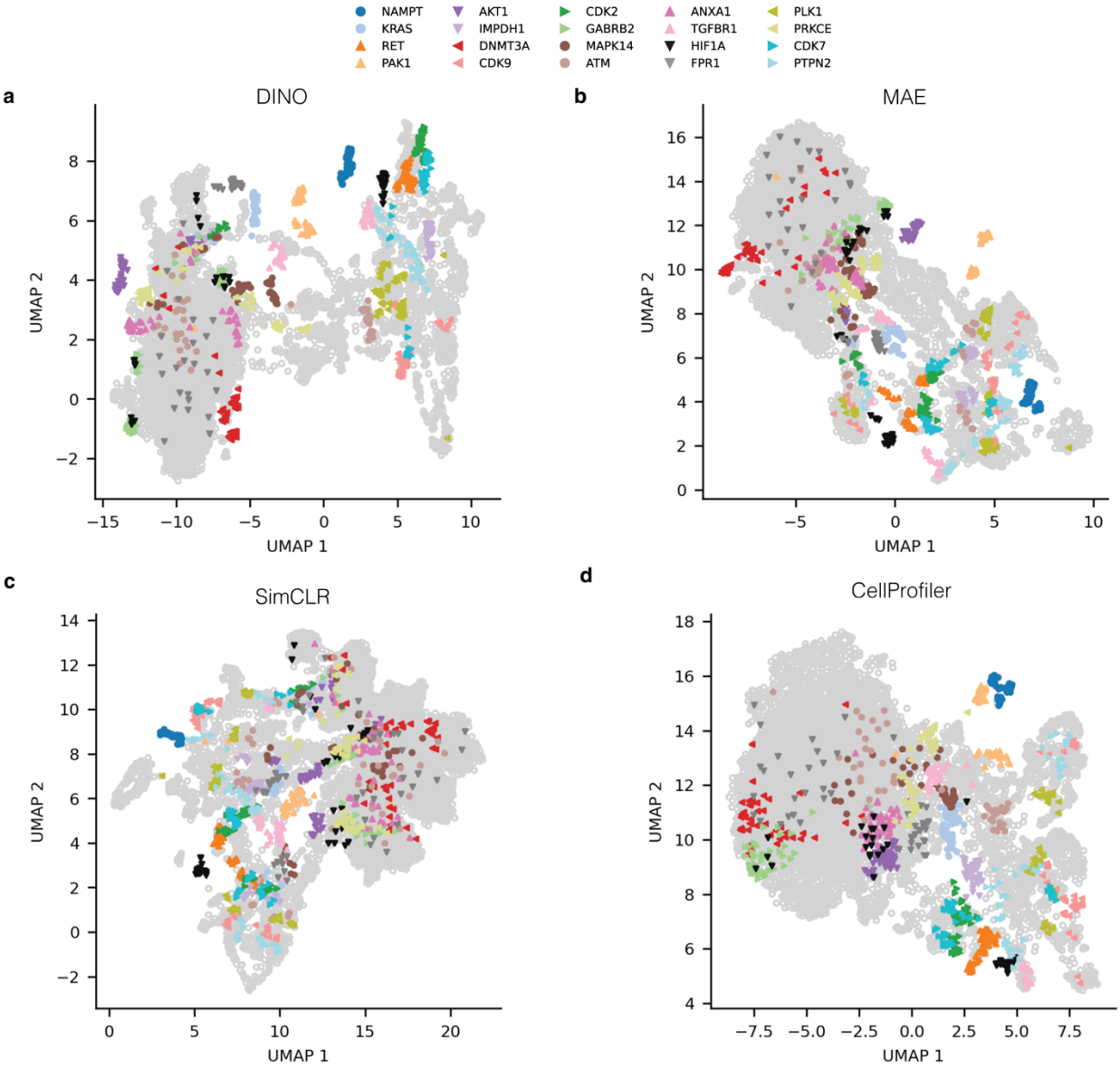

Figure 4: UMAP projections of SSL and CellProfiler representations reveal compound clusters. Two-dimensional projections of well-level representations from SSL methods and the CellProfiler baseline on the multisource evaluation set. Colors highlight target labels for the set of targets with the highest F1-scores across all methods. Perturbations with other targets are depicted as grey hollow points. All SSL models used the ViT-S architecture.

We first qualitatively assessed the biological relevance of representations by highlighting the top 20 most predictable targets (see Methods; Fig.4). All representations grouped compounds with the same target to some extent, with DINO and CellProfiler producing well-separated clusters. This is remarkable given that the well profiles originate from different batches and laboratories. We observed striking similarities between the UMAP projections of DINO and CellProfiler representations. For instance, both DINO and CellProfiler placed CDK2, CDK7, RET, HIF1A, and TGBR1 targeting compounds in close proximity. These results further support that DINO representations are biologically meaningful and at least as powerful as the handcrafted CellProfiler features.

We then examined the robustness of representations with respect to technical sources of variation by coloring UMAP projections by batch and source (Supplementary Fig. 7a,b). SSL features were more susceptible to technical variations compared to CellProfiler, displaying a stronger separation between experimental subgroups. These results contrast with the findings from quantitative metrics, where DINO outperformed CellProfiler in matching perturbations across batches (Fig. 3a,c) and sources (Supplementary Fig. 6e). Upon closer inspection, we found that the most pronounced laboratory effects occurred within DMSO negative controls (Supplementary Fig. 7a,c) and that the differences strongly correlated with variations in cell count (Supplementary Fig. 7d). We hypothesize that CellProfiler is more robust towards cell count variations since the features are extracted from single cells.

Additionally, we quantified the effect of technical variations using 3 metrics: *Silhouette score*[49] for group separability, *Graph Connectivity (GC)*[50] for group connectivity and *Local Inverse Simpson's Index (LISI)*[50] for neighborhood diversity. Higher *Silhouette* and *GC* values imply stronger group separation and therefore stronger source/batch effects, while higher *LISI* values indicate milder effects (see Methods). In general, all representations were affected by batch and source effects (Supplementary Fig. 8), being closer to non-overlapping Gaussian distributions than a randomized baseline (see Methods). CellProfiler was the least sensitive to technical biases, although DINO showed the lowest *source-GC* score (Supplementary Fig. 8d). Together, these results suggest that SSL representations capture both biological and technical axes of variation in the data.

Hierarchical clustering of DINO representations recapitulates known gene families

Finally, we assessed the generalizability of SSL models on an independent dataset consisting of gene overexpression perturbations (ORFs) imaged at a new source (see “JUMP-CP datasets for benchmarking SSL representations”). Since ORF perturbations refer to individual genes, we used *gene family* annotations from the HUGO Gene Nomenclature Committee (HGNC) as labels to compute metrics of biological relevance.

The evaluation of SSL models on the gene overexpression set provided further evidence that DINO learned the most generalizable representations, thereby outperforming CellProfiler (Fig. 5a-c). As for the chemical perturbations, DINO demonstrated the best reproducibility (Fig. 5a) and biological relevance (Fig. 5b) among all models. The most significant improvement was observed in DINO's ability to predict gene family labels using the nearest neighbor classifier (*NSBP accuracy*), with a relative accuracy gain of 64% compared to CellProfiler. As illustrated in Fig. 5c, DINO consistently outperformed CellProfiler in classifying gene families for a wide

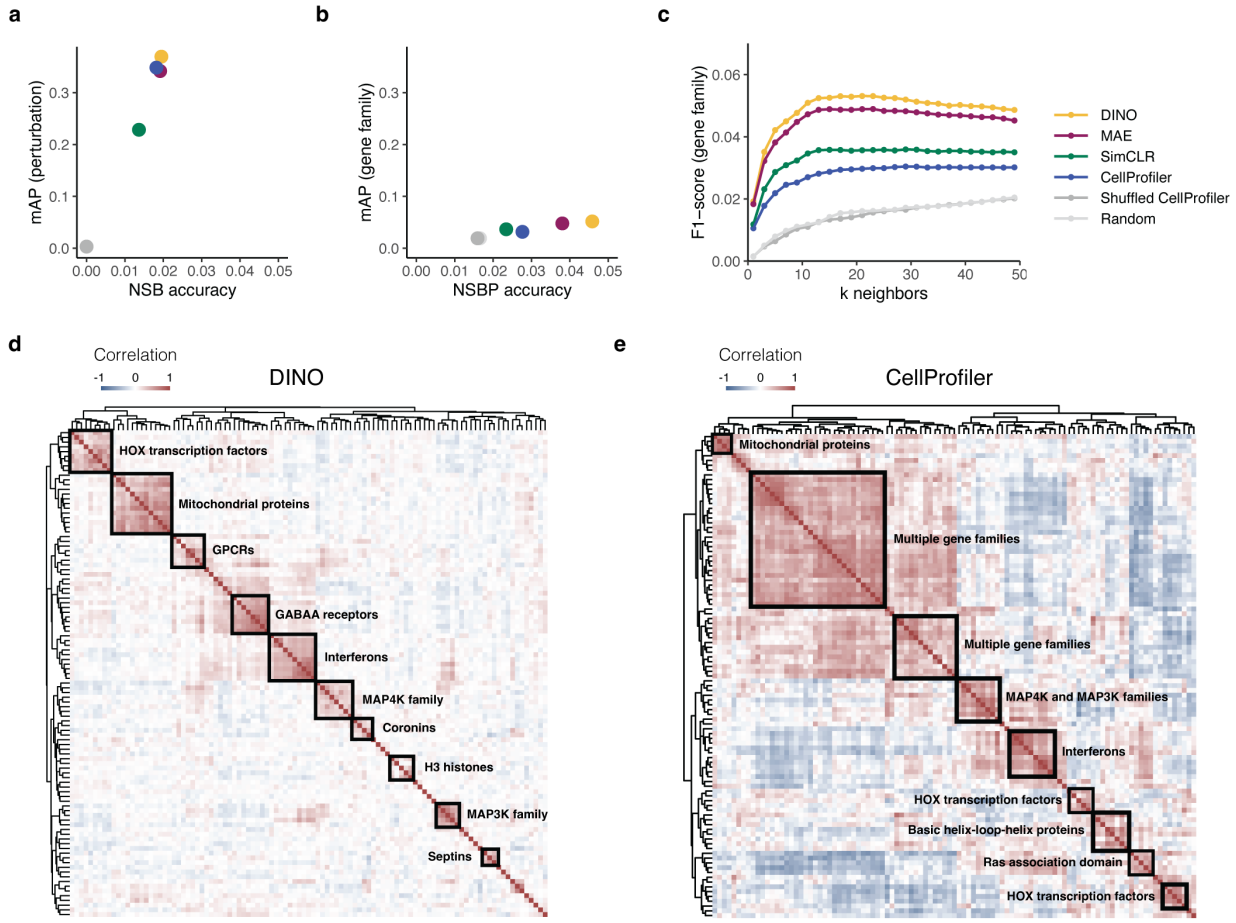

Figure 5: Evaluation of SSL models on the gene overexpression data demonstrates generalizability to unseen data.
Evaluation of SSL models and CellProfiler on an independent gene overexpression set (see Methods). SSL models were trained on the multisource dataset with the ViT-S architecture. Colors indicate different models and two randomized baselines: Shuffled CellProfiler (CellProfiler features with shuffled labels) and Random (random normally distributed features). **a)** Reproducibility metrics: *perturbation not-same-batch (NSB) accuracy* and *mean average precision (mAP)*. **b)** Metrics of biological relevance: *gene family not-same-batch-or-perturbation (NSBP) accuracy* and *mean average precision (mAP).* **c)** F1-scores for matching gene family labels based on ORF consensus profiles for a range of nearest neighbors *k.* **d)-e)** Hierarchical clustering of the 20 gene families with the highest intragroup correlations in the DINO and CellProfiler representation spaces, respectively. Detailed versions of the heatmaps displaying gene and gene family annotations for each row are presented in Supplementary Fig. 9-10.

range of *k* nearest neighbors in terms of F1-score. In general, the performance metrics were much lower for ORF perturbations than for chemical perturbations, consistent with the previously reported lower reproducibility of ORFs over compounds[35].

To further investigate the expressiveness of the learned representations, we evaluated their ability to recapitulate gene families in an unsupervised manner using clustering. For each representation type, we selected the 20 gene families with the highest within-family correlations (see Methods) and performed hierarchical clustering of ORF consensus profiles for

DINO (Fig. 5d) and CellProfiler (Fig. 5e). The resulting heatmaps were annotated to highlight gene groups based on gene and gene family labels (see Methods).

Our qualitative results from similarity heatmaps revealed that DINO representations recovered a larger number of gene groups and produced more homogeneous clusters than CellProfiler. DINO recapitulated 10 gene groups (Fig. 5d and Supplementary Fig. 9), including HOX transcription factors, mitochondrial proteins, GPCRs, GABAA receptors, interferons, MAP4K family, coronins, H3 histones, MAP3K family, and septins. By contrast, CellProfiler recovered only 6 gene groups (mitochondrial proteins, MAP4K/MAP3K family, interferons, HOX transcription factors, basic helix-loop-helix proteins, and Ras association domain) and two large clusters of mixed gene families (Fig. 5e and Supplementary Fig. 10). To provide a more objective assessment, we defined 20 clusters using agglomerative clustering and computed the adjusted mutual information (AMI) between cluster assignments and gene family labels (see Methods). We found that DINO (AMI = 0.50) outperformed CellProfiler (AMI = 0.22) on the gene clustering task. Taken together, these findings demonstrate the applicability of self-supervised methods for analyzing multichannel microscopy images as well as their transferability across domain-relevant distribution shifts such as perturbation types (chemical vs genetic).

SSL pipeline is significantly faster than CellProfiler

Finally, we compared the computation time and cloud costs of performing feature extraction using DINO and CellProfiler. For this benchmark, we used 12 GPU-accelerated cloud instances for DINO and 12 CPU-intensive instances for CellProfiler (Supplementary Table 1), analyzing 28 plates with each method. We found that DINO was 50 times faster than CellProfiler, with an average processing time of 1.3 minutes per plate (Supplementary Table 1). Despite the need for GPU resources, the cloud costs per plate were over 50 times lower for DINO than for CellProfiler (Supplementary Table 1). Moreover, DINO offered a simple workflow which processed images end-to-end, in contrast to the multi-step Cell Profiler workflow which requires illumination correction, segmentation, and feature extraction.

Together, our findings show that SSL methods are a viable alternative to traditional segmentation-based approaches, offering improved performance, speed, and lower workflow complexity and computational costs.

Discussion

Our comprehensive benchmark study demonstrates the applicability of SSL methods for extracting information-rich and transferable features from Cell Painting images. Besides providing powerful representation learning models, our work offers practical guidance for SSL applications in the multichannel microscopy image domain.

We systematically examined the role of augmentations in joint-embedding methods in the context of Cell Paining. Our results identified the use of 'Color' augmentation as the most crucial. Without this augmentation, the pretext task becomes trivial and can be “solved” through shortcut learning[51], by matching the low-level statistics (e.g. color histograms) as previously shown for natural images[15]. We observed, however, that the 'Resize' augmentation reduced representational power. We hypothesize that enforcing scale-invariance is detrimental since microscopy images are typically acquired with the same optical parameters, with cell size

variations reflecting actual phenotypic changes. This emphasizes the need for careful reconsideration of design choices when adapting methods from the natural image domain to microscopy images.

The size and heterogeneity of the training set played a crucial role for SSL model performance. We observed that scaling up from a single-source to a multisource training set led to a significant increase in performance, even more than using larger ViT architectures, which may require bigger training sets[52]. Interestingly, our SSL models performed well on an independent gene overexpression set without fine-tuning, despite being trained only on compound perturbations. This suggests the potential for SSL methods to generalize to unseen perturbations and biological datasets. On the other hand, the low performance of our transfer learning baseline suggests that pretrained ImageNet weights cannot be directly utilized without domain-specific fine-tuning.

Previous studies[26,28,32] have used compound treatment label prediction or adapted ImageNet weights to learn domain-relevant representations for Cell Painting. A drawback of treatment prediction approaches is their reliance on compound preselection to construct a curated training set, as opposed to our end-to-end SSL workflow. Simultaneously, the adoption of pretrained weights from natural images comes with the limitation that each channel is typically fed independently to the network followed by concatenation of single channel representations. Consequently, the computational time complexity of these approaches scales linearly with the number of input channels and results in high-dimensional (> 1000) features. By contrast, our SSL models were trained using architectures tailored for 5-channel images, facilitating joint channel learning, and avoiding concatenation. As a result, SSL representations exhibited reduced redundancy, with only one feature removed from DINO and MAE representations, when applying correlation-threshold-based feature selection.

In contrast to previous works that only focused on smaller scale Cell Painting datasets (BBBC022[53], BBBC036[54], BBBC037[55], BBBC047[54]), we are the first to present a holistic self-supervised study that leverages the JUMP-CP consortium data[35]. Still, we are only using a subset of the full JUMP-CP dataset. The reason behind this is that with our current hardware (NVIDIA Tesla V100 with 32GB of VRAM), training on the data subset already requires approximately 670 and 1340 GPU hours for training DINO ViT-S and ViT-B, which translates to 1 and 2 weeks using distributed training with 4 GPUs, respectively.

One limitation of our work is that since SSL models do not operate on the single-cell level, it is not possible to argue about cell heterogeneity. In such scenarios where cell-based analysis is desirable, we believe that CellProfiler is still the dominant tool. CellProfiler features are also more interpretable, tracing back to specific microscopy channels and morphological descriptors. Nevertheless, by utilizing transformer architectures, our models allow interpretation of relevant image regions through the inspection of self-attention maps[16].

Finally, our analysis showed that self-supervised representations are more susceptible than CellProfiler to technical effects and variations in cell count, possibly because they use multicellular image crops instead of single cells. The design of strategies to achieve batch and source invariance in SSL representations remains an open question. Another promising research direction is to study the scaling behavior of larger ViTs using the full JUMP-CP

dataset. Similar to advancements in natural image pretraining, large-scale pretraining can either be achieved by visual self-supervision or by contrastive language-image pretraining. An interesting work in this direction was recently presented in [56], using chemical text-based representations such as SMILES jointly with Cell Painting images. We encourage future work to explore this in greater depth.

Methods

JUMP-CP single-source and multisource data

We used the dataset cpg0016-jump[35], available from the Cell Painting Gallery on the Registry of Open Data on AWS (https://registry.opendata.aws/cellpainting-gallery/) for training and evaluation of self-supervised learning (SSL) models. All images used in the study were 5-channel Cell Painting images obtained with a 20X objective. The single-source training set consists of 391,815 images from JUMP-CP source 3, corresponding to 35,892 compounds with 1 replicate and 9 fields of view per well. The multisource training set contains 564,272 images from 4 JUMP-CP sources: source 2, source 3, source 6, and source 8. The multisource training set includes 5 replicates of 10,057 compounds from Selleckchem and MedChemExpress bioactive libraries, with two replicates originating from source 3. The SSL models were evaluated using JUMP-Target2 plates[35] with 302 target-annotated compounds. Single-source (6 batches, 33,962 images) and multi-source (16 batches, 75,545 images) evaluation sets were constructed from JUMP-Target2 plates obtained from the corresponding sources and screening batches.

Gene overexpression data

The JUMP-CP[35] gene overexpression data (source 4) with open reading frames (ORFs) was used for the final assessment of self-supervised learning (SSL) models. A subset of the gene overexpression data was constructed by selecting ORF perturbations with high replicate correlations ($r > 0.4$) in the CellProfiler feature space, resulting in 4,339 ORFs. Gene group memberships were assigned to each ORF perturbation using the HGNC gene annotation (hgnc_complete_set_2022-10-01.txt). To ensure robust evaluation based on gene annotations, we selected only those gene groups with at least 4 unique ORFs. 1,466 ORF perturbations satisfied this criterion.

Image preprocessing

The JUMP-CP consortium[35] generated Cell Painting images with five channels (Mito, AGP, RNA, ER, DNA) that were stored as individual TIFF files. To optimize data loading, we combined the single-channel images into 5-channel TIFF files, resulting in a 6-fold acceleration in training time. Prior to storage, we preprocessed the images: for each channel, intensities were clipped at 0.01st and 99.9th percentiles and scaled to the range [0,1]. Additionally, we calculated the Otsu threshold in the DNA staining channel and saved it as image metadata. During training, this threshold value enabled us to sample non-empty crops based on the minimum percentage of foreground area in the DNA channel.

Augmentations for multichannel images

In the augmentation study, we tested 6 augmentations: 'Flip', 'Resize', 'Color', 'Drop channel', 'Gaussian noise' and 'Gaussian blur'. The 'Flip' augmentation rotates an image by 180 degrees along the horizontal or vertical axes. 'Resize' generates crops with dimensions varying between 12% and 47% of the whole microscopy image and rescales the output to 224x224 pixels. The 'Color' augmentation consists of a random intensity shift: $I_{ch} = I_{ch} + \varepsilon,\ \varepsilon \in U(-0.3, 0.3)$ and a random brightness change: $I_{ch} = I_{ch}^{\gamma},\ \gamma \in U(0.5, 1.5)$ with intensity values restricted to $[0, 1]$. 'Drop channel' omits one channel from the image at random with probability $p = 0.5$. The dropped channel is padded with zeros. 'Gaussian noise' adds random noise to the image: $I_{ch} = I_{ch} + \mu,\ \mu \in N(0, 0.05)$. 'Gaussian blur' applies a Gaussian filter with a kernel size of 23 and a standard deviation uniformly sampled from $[0.1, 2]$. Based on the results of the augmentation study, we applied only 'Flip' and 'Color' augmentations for training SimCLR and DINO. For training MAE, only the 'Flip' augmentation was used.

Model training details

During training, we sampled random crops (224x224) from the images and provided these as inputs to the models. For the joint-embedding models SimCLR and DINO, we additionally applied an augmentation pipeline, as described in "Augmentations for multichannel images", to generate multiple views from the sampled crops. All input crops were centered and scaled using channel intensity means and standard deviations estimated over the entire training set. We used the small (ViT-S/16) and base (ViT-B/16) variants of the vision transformer, with a patch size of 16. The models with ViT-S/16 were trained for 200 epochs, while those with ViT-B/16 were trained for 400 epochs. We used the AdamW optimizer and saved checkpoints every 20 epochs. In addition to tracking the SSL training loss, which can be an unreliable indicator of downstream performance, we monitored training progress using the mean replicate correlation on the single-source evaluation set and selected the best-performing checkpoint for each model. A brief exposition of model-specific hyperparameters is provided below. For a comprehensive overview of training hyperparameters refer to Supplementary Table 2.

DINO was trained with a batch size of 128 (192 for ViT-B), a learning rate of $2 \cdot 10^{-3}$ ($1.5 \cdot 10^{-3}$ for ViT-B), and a weight decay linearly increasing from 0.04 to 0.4. The learning rate followed a 20-epoch linear warmup followed by a cosine decay. For each image, 8 local crops (96x96) and 2 global crops (224x224) were sampled. DINO is a joint-embedding model with a student-teacher architecture[16]. DINO projects representations into a high-dimensional (here 20,000-dimensional) space where the temperature-scaled cross-entropy loss is optimized using a temperature of 0.1 for the student and 0.04 for the teacher network. The teacher temperature followed a linear warmup starting from 0.01 for 30 epochs.

Masked autoencoder (MAE) was trained with a batch size of 1024 (1536 for ViT-B), a learning rate of $6 \cdot 10^{-4}$ ($9 \cdot 10^{-4}$ for ViT-B), and a weight decay of 0.05. The learning rate followed a 30-epoch linear warmup followed by a cosine decay. Given a partially masked input image, MAE reconstructs the missing regions using an asymmetric encoder-decoder architecture[17], with a significantly smaller decoder. To accelerate data loading, 4 random crops were sampled from each image during training. The masking ratio was set to 50%, and image augmentation was performed using only horizontal and vertical flips.

SimCLR was trained with a batch size of 256, a learning rate of $1 \cdot 10^{-3}$, and a weight decay of 0.1. The learning rate followed a 30-epoch linear warmup followed by a cosine decay. SimCLR is a contrastive approach[15] that aims to match augmented views from the same image in the representation space ("positives"), while pushing away representations from different images ("negatives"). The temperature-scaled cross-entropy loss was used as the objective function with a constant temperature value of 0.2.

SSL inference and postprocessing

Each field-of-view (FOV) image was split into 224x224 image crops $\{x_i, i = 1, \dots, N_{\text{crops}}\}$, which were passed through a pretrained backbone of DINO/SimCLR or a pretrained encoder of MAE. We only used image crops with cells ("cell-centered crops"), which was ensured by imposing a lower bound of 1% on the Otsu-thresholded area in the DNA channel. The resulting image crop representations were aggregated using the arithmetic mean: $f_{\text{img}} = \frac{1}{N_{\text{crops}}} \sum_{i=1}^{N_{\text{crops}}} f(x_i)$. Well representations were obtained by taking the mean across all FOV images: $f_{\text{well}} = \frac{1}{N_{\text{FOV}}} \sum_{k=1}^{N_{\text{FOV}}} f_{\text{img}}^{k}$. Features with variance $< 1 \cdot 10^{-5}$ were removed.

Sphering transformation[22] Φ was applied to the well representations, followed by normalization using whole-plate median (med) and median absolute deviation (MAD):

$$f_{\text{norm}} = \frac{\Phi(f_{\text{well}}) - med\big(\Phi(f_{\text{well}})\big)}{MAD(\Phi(f_{\text{well}}))}$$

Consensus perturbation profiles were obtained by taking the mean across normalized well representations of screened replicates: $f_{\text{cons}} = \frac{1}{N_{\text{rep}}} \sum_{i=1}^{N_{\text{rep}}} f_{\text{norm}}^{i}$. Batch-aggregated perturbation profiles were estimated by taking the mean across normalized well representations of the same perturbation within each batch. Source-aggregated profiles were estimated by taking the mean of normalized well representations from all wells and batches within each JUMP-CP source.

CellProfiler features

We used CellProfiler features provided by the JUMP-CP consortium (https://registry.opendata.aws/cellpainting-gallery/, for details see [35]). CellProfiler features were normalized using whole-plate median and MAD ("MAD robustize" normalization), which was determined to be the best postprocessing method for CellProfiler features (Supplementary Fig. 2e). We used the set of 560 features selected in the CPJUMP1 study[47], in which low-variance and redundant features were removed based on a dataset with chemical and genetic (ORF and CRISPR) perturbations.

Transfer learning

A vision transformer (ViT-S/16 or ViT-B/16) pretrained on the image classification task on ImageNet-1K was used to extract 'transfer learning' representations. Each of the 5 channels was duplicated 3 times to generate pseudo-RGB images, which were individually passed through a pretrained architecture. The transfer learning representations were obtained by concatenating individual channel representations, resulting in $5 \cdot 384 = 1920$-dimensional feature vectors. As for SSL representations, low variance features ($< 1 \cdot 10^{-5}$) were removed

before normalization. The transfer learning representations were normalized using whole-plate median and MAD ("MAD robustize" normalization), which was the best postprocessing method for transfer learning (Supplementary Fig. 2d).

Evaluation of representations

Mean average precision (mAP) was used to assess the performance of k-nearest neighbor (kNN) classification on both perturbation and target labels, with *perturbation mAP* quantifying reproducibility and *target mAP* quantifying biological relevance of representations. To compute *perturbation mAP*, *k* nearest neighbors of each batch-aggregated profile were queried in the representation space, and the number of neighbors with matching perturbation labels (true positives, TP) and those with non-matching labels (false positives, FP) were counted. True negatives (TN) and false negatives (FN) were determined by considering profiles outside of the defined *k*-neighborhood. To aggregate precision and recall values, average precision (AP) was calculated by varying the number of nearest neighbors *k*. Mean average precision (mAP) was obtained by aggregating AP values across all batch-aggregated profiles. *Target mAP* was estimated similarly to *perturbation mAP* but with the target label to determine precision-recall and using consensus profiles instead of batch-aggregated profiles. For ORF perturbations, the biological relevance of representations was estimated using *gene family mAP* which was calculated on consensus profiles using the gene family label. To evaluate the cross-source matching ability of representations (Supplementary Fig. 6e), we calculated *perturbation mAP* and *not-same-source (NSS) accuracy* on source-aggregated profiles.

In addition to AP values, F1-scores at *k* nearest neighbors were computed and visualized to investigate whether some representations worked better in a specific *k* range. We also considered several alternatives to *mAP*, such as *mean precision at R* and *area under micro-averaged precision-recall curves* (*AUCPR)*, which produced similar results (see Supplementary Fig. 4 for details).

NSB accuracy and *NSBP accuracy* were estimated on well profiles. To calculate these metrics, pairwise distances were computed between all samples in the representation space, and for each sample, the nearest neighbor from a different batch (NSB) or with a different perturbation label (NSBP) was identified. *NSB accuracy* was calculated by comparing the perturbation labels of the query and the nearest neighbor, while *NSBP accuracy* was calculated by comparing the target labels. In both cases, the accuracy reflects the percentage of correct predictions.

UMAP projections

To visualize our representations in 2 dimensions, we provided the first 200 principal components (PCs) as input to generate UMAP[48] projections with the correlation distance metric used to compute distances between points. To ensure optimal visualization, we set the number of nearest neighbors to 50 and the minimum distance between points to 0.7 in the UMAP algorithm.

We used the JUMP-Target2 plate[35] target annotation and selected 20 drug targets with the highest mean F1-scores, which were determined by performing kNN target classification using CellProfiler and DINO representations. Supplementary Table 3 provides the list of these 20 targets and their F1-scores stratified by representation type.

Quantification of batch and laboratory effects

Following [50], we used *silhouette scores, graph connectivity (GC)*, and *local inverse Simpson's index* (*LISI)* to quantify batch and laboratory effects based on well profiles. For this analysis, group labels were assigned using the true batch and source labels. A silhouette score measures the similarity of an observation to its own group relative to the nearest group[49].

Both *GC* and *LISI* are based on a kNN-graph *G(V, E)*, which consists of a set of vertices *V* corresponding to the well profiles. Each profile is connected to its closest *k* neighbors with respect to the cosine distance, thus defining the edge set *E*. Let *S* be a set of categories, such as batches or laboratories. Taking only the well profiles based on a specific category $s \in S$ induces a subgraph $G_s(V_s, E_s)$. GC measures on average the ratio between the number of profiles in the largest connected component *(LCC*) of G_s and the total number of nodes in the subgraph across all categories in *S*:

$$GC = \frac{1}{S} \sum_{s \in S} \frac{|LCC(G_s(V_s, E_s))|}{|V_s|}.$$

If the LCC of G_s is almost as large as G_s itself, this indicates that profiles from the same category are close to each other – a clear sign of batch effects. We reported GC for *k* = 1, 2, 3, 5, 10, 15.

LISI quantifies neighborhood diversity in the kNN-graph *G* and is calculated based on the inverse Simpson's index

$$LISI = \frac{1}{\sum_{s \in S} p(s)}.$$

LISI can be interpreted as the expected number of profiles that need to be sampled, before two profiles are drawn from the same category[57]. A higher *LISI* implies a more diverse neighborhood, and thus lower batch effects. LISI was calculated for *k* = 15, 30, 60, 90. For both *GC* and *LISI*, the values for *k* are based on [50].

For the sake of interpretability, we incorporated two baselines: 1) a Gaussian baseline, which simulates non-overlapping Gaussian ($\sigma = 1$) clusters with the number of clusters equal to the number of batches/sources and with feature dimensionality identical to that of CellProfiler representations; 2) a random baseline, which corresponds to the Gaussian baseline but with randomized cluster assignments.

Hierarchical clustering of genetic perturbations

For the clustering analysis, we only considered HGNC gene families (see “Gene overexpression data”) that contained at least 4 unique ORF perturbations. Since many of the HGNC gene families were heterogeneous and uncorrelated, we only selected the top 20 gene families with the highest within-family correlations in the respective representation space, resulting in 2 gene sets for DINO and CellProfiler.

To cluster these gene sets, we calculated the gene-gene correlation matrix within the respective representation space, which was then provided as input for hierarchical clustering using the complete linkage method and the Euclidean distance metric. To highlight biologically meaningful

clusters in box frames, gene groups with at least 3 perturbations were identified visually and labeled with the majority gene family label.

To evaluate the quality of hierarchical clustering of genetic perturbations, we used *adjusted mutual information (AMI)*. For a given number of clusters, mutual information (MI) quantifies the dependence between cluster assignment labels X and gene family labels Y:

$$MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} P_{(X,Y)}(x,y) \log\left(\frac{P_{(X,Y)}(x,y)}{P_X(x)P_Y(y)}\right)$$

Adjusting mutual information (MI) for random chance results in AMI:

$$AMI(X,Y) = \frac{MI(X,Y) - E\{MI(X,Y)\}}{\max\{H(X), H(Y)\} - E\{MI(X,Y)\}}$$

Implementation details

All self-supervised learning (SSL) models were implemented in Python 3.9.7 using PyTorch[58] v1.10.2 and PyTorch Lightning v1.6.3. SSL model training and inference were conducted on Tesla V100 GPUs with a VRAM of 32GB. The postprocessing of representations (sphering, MAD robustize, standardize) was carried out using pycytominer v0.2.0. Mean average precision (mAP), NSB and NSBP accuracies were computed using custom Python functions, with average precision (AP) and accuracy scores computed using the scikit-learn[59] v1.0.2 implementation. PCA and UMAP were performed using scikit-learn v1.0.2 and umap-learn v0.5.3. The silhouette scores, kNN graphs, and the simulated Gaussian baseline for batch and source effect quantification were computed using scikit-learn v1.0.2. The LISI scores were calculated using HarmonyPy[57] v0.0.9. Adjusted mutual information (AMI) of gene family clustering was calculated using scikit-learn v1.0.2. The visualization of UMAP was performed using matplotlib v3.5.0 and seaborn v0.11.2. The visualization of evaluation metrics and hierarchical clustering of ORF perturbations were performed in R 4.1.2.

Software availability

The code for training, inference, and evaluation of the self-supervised learning (SSL) models used in this study will be made publicly available on GitHub upon publication of this article in a journal.

Acknowledgements

We would like to express our gratitude to the following individuals and organizations for their contributions to this work. We thank Bayer AG, which enabled the successful completion of this work through the Life Science Collaboration (LSC) project 'Picasso'. We thank all members of the JUMP-CP consortium for producing the Cell Painting dataset that was used for training SSL models in this study. We are particularly grateful to Shantanu Singh, Niranj Chandrasekaran and John Arevalo from the Broad Institute for their data curation efforts within the JUMP-CP consortium. We would like to thank Djork-Arne Clevert for his support and guidance throughout this project. Finally, we thank all members of the Machine Learning Research group at Bayer

AG for their kind advice and David Gnutt and Tim Koenig for their valuable feedback on our scientific work.

Author contributions

VK and PAMZ designed the study. PAMZ supervised the study. VK prepared training and evaluation data. VK and NA implemented the self-supervised learning (SSL) framework including data loading, augmentation, and training pipelines. VK and NA trained SSL models. MO and VK designed and implemented a cloud-based inference pipeline. VK and MO produced representations for the gene overexpression data using the inference pipeline. VK conducted an augmentation study. VK and NA performed evaluations of SSL representations. FM conducted batch and laboratory effect analysis on the SSL representations. VK designed the figures, with contributions from PAMZ, MO, FM and NA. VK, NA and PAMZ wrote the manuscript with inputs from MO and FM. All authors reviewed and approved the final version.

Conflict of interest

The authors declare that they have no conflicts of interest related to this publication.

References

1. Boutros, M., Heigwer, F. & Laufer, C. Microscopy-Based High-Content Screening. *Cell* **163**, 1314–1325 (2015).
2. Loo, L.-H., Wu, L. F. & Altschuler, S. J. Image-based multivariate profiling of drug responses from single cells. *Nat. Methods* **4**, 445–453 (2007).
3. Carpenter, A. E. Image-based chemical screening. *Nat. Chem. Biol.* **3**, 461–465 (2007).
4. Mattiazzi Usaj, M. *et al.* High-Content Screening for Quantitative Cell Biology. *Trends Cell Biol.* **26**, 598–611 (2016).
5. Reisen, F. *et al.* Linking Phenotypes and Modes of Action Through High-Content Screen Fingerprints. *ASSAY Drug Dev. Technol.* **13**, 415–427 (2015).
6. Ziegler, S., Sievers, S. & Waldmann, H. Morphological profiling of small molecules. *Cell Chem. Biol.* **28**, 300–319 (2021).
7. MacDonald, M. L. *et al.* Identifying off-target effects and hidden phenotypes of drugs in human cells. *Nat. Chem. Biol.* **2**, 329–337 (2006).

8. Chow, Y. L., Singh, S., Carpenter, A. E. & Way, G. P. Predicting drug polypharmacology from cell morphology readouts using variational autoencoder latent space arithmetic. *PLoS Comput. Biol.* **18**, e1009888 (2022).
9. Jin, G. & Wong, S. T. C. Toward better drug repositioning: prioritizing and integrating existing methods into efficient pipelines. *Drug Discov. Today* **19**, 637–644 (2014).
10. Karaman, B. & Sippl, W. Computational Drug Repurposing: Current Trends. *Curr. Med. Chem.* **26**, 5389–5409 (2019).
11. Mirabelli, C. *et al.* Morphological cell profiling of SARS-CoV-2 infection identifies drug repurposing candidates for COVID-19. *Proc. Natl. Acad. Sci.* **118**, e2105815118 (2021).
12. Nyffeler, J. *et al.* Bioactivity screening of environmental chemicals using imaging-based high-throughput phenotypic profiling. *Toxicol. Appl. Pharmacol.* **389**, 114876 (2020).
13. 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).
14. Jing, L. & Tian, Y. Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 4037–4058 (2021).
15. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. in *Proceedings of the 37th International Conference on Machine Learning* 1597–1607 (PMLR, 2020).
16. Caron, M. *et al.* Emerging Properties in Self-Supervised Vision Transformers. in 9650–9660 (2021).
17. He, K. *et al.* Masked Autoencoders Are Scalable Vision Learners. in 16000–16009 (2022).
18. McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology. *PLOS Biol.* **16**, e2005970 (2018).
19. Berg, S. *et al.* ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* **16**, 1226–1232 (2019).

20. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).

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

22. Ando, D. M., McLean, C. Y. & Berndl, M. Improving Phenotypic Measurements in High-Content Imaging Screens. 161422 Preprint at https://doi.org/10.1101/161422 (2017).

23. 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).

24. Lafarge, M. W. *et al.* Capturing Single-Cell Phenotypic Variation via Unsupervised Representation Learning. in *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning* 315–325 (PMLR, 2019).

25. Dürr, O. & Sick, B. Single-Cell Phenotype Classification Using Deep Convolutional Neural Networks. *J. Biomol. Screen.* **21**, 998–1003 (2016).

26. 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).

27. Le, T. *et al.* Analysis of the Human Protein Atlas Weakly Supervised Single-Cell Classification competition. *Nat. Methods* **19**, 1221–1229 (2022).

28. Moshkov, N. *et al.* Learning representations for image-based profiling of perturbations. 2022.08.12.503783 Preprint at https://doi.org/10.1101/2022.08.12.503783 (2022).

29. Kraus, O. Z., Ba, J. L. & Frey, B. J. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinforma. Oxf. Engl.* **32**, i52–i59 (2016).

30. Godinez, W. J., Hossain, I., Lazic, S. E., Davies, J. W. & Zhang, X. A multi-scale convolutional neural network for phenotyping high-content cellular images. *Bioinforma. Oxf. Engl.* **33**, 2010–2019 (2017).

31. Pawlowski, N., Caicedo, J. C., Singh, S., Carpenter, A. E. & Storkey, A. Automating Morphological Profiling with Generic Deep Convolutional Networks. 085118 Preprint at https://doi.org/10.1101/085118 (2016).
32. Cross-Zamirski, J. O. *et al.* Self-Supervised Learning of Phenotypic Representations from Cell Images with Weak Labels. Preprint at https://doi.org/10.48550/arXiv.2209.07819 (2022).
33. Kobayashi, H., Cheveralls, K. C., Leonetti, M. D. & Royer, L. A. Self-supervised deep learning encodes high-resolution features of protein subcellular localization. *Nat. Methods* **19**, 995–1003 (2022).
34. Janssens, R., Zhang, X., Kauffmann, A., de Weck, A. & Durand, E. Y. Fully unsupervised deep mode of action learning for phenotyping high-content cellular images. *Bioinforma. Oxf. Engl.* btab497 (2021) doi:10.1093/bioinformatics/btab497.
35. Chandrasekaran, S. N. *et al.* JUMP Cell Painting dataset: morphological impact of 136,000 chemical and genetic perturbations. 2023.03.23.534023 Preprint at https://doi.org/10.1101/2023.03.23.534023 (2023).
36. Gidaris, S., Singh, P. & Komodakis, N. Unsupervised Representation Learning by Predicting Image Rotations. Preprint at https://doi.org/10.48550/arXiv.1803.07728 (2018).
37. Noroozi, M. & Favaro, P. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. in *Computer Vision – ECCV 2016* (eds. Leibe, B., Matas, J., Sebe, N. & Welling, M.) 69–84 (Springer International Publishing, 2016). doi:10.1007/978-3-319-46466-4_5.
38. Misra, I. & Maaten, L. van der. Self-Supervised Learning of Pretext-Invariant Representations. in 6707–6717 (2020).
39. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. Preprint at https://doi.org/10.48550/arXiv.1911.05722 (2020).

40. Grill, J.-B. *et al.* Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. in *Advances in Neural Information Processing Systems* vol. 33 21271–21284 (Curran Associates, Inc., 2020).
41. Chen, R. J. & Krishnan, R. G. Self-Supervised Vision Transformers Learn Visual Concepts in Histopathology. Preprint at https://doi.org/10.48550/arXiv.2203.00585 (2022).
42. Xie, Y., Zhang, J., Xia, Y. & Wu, Q. UniMiSS: Universal Medical Self-supervised Learning via Breaking Dimensionality Barrier. in *Computer Vision – ECCV 2022* (eds. Avidan, S., Brostow, G., Cissé, M., Farinella, G. M. & Hassner, T.) 558–575 (Springer Nature Switzerland, 2022). doi:10.1007/978-3-031-19803-8_33.
43. Dosovitskiy, A. *et al.* An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Preprint at https://doi.org/10.48550/arXiv.2010.11929 (2021).
44. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems* vol. 30 (Curran Associates, Inc., 2017).
45. Assran, M. *et al.* Masked Siamese Networks for Label-Efficient Learning. in *Computer Vision – ECCV 2022* (eds. Avidan, S., Brostow, G., Cissé, M., Farinella, G. M. & Hassner, T.) 456–473 (Springer Nature Switzerland, 2022). doi:10.1007/978-3-031-19821-2_26.
46. Dehghani, M. *et al.* Scaling Vision Transformers to 22 Billion Parameters. Preprint at http://arxiv.org/abs/2302.05442 (2023).
47. Chandrasekaran, S. N. *et al.* Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. 2022.01.05.475090 Preprint at https://doi.org/10.1101/2022.01.05.475090 (2022).
48. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Preprint at https://doi.org/10.48550/arXiv.1802.03426 (2020).

49. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).

50. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).

51. Geirhos, R. *et al.* Shortcut Learning in Deep Neural Networks. *Nat. Mach. Intell.* **2**, 665–673 (2020).

52. Zhai, X., Kolesnikov, A., Houlsby, N. & Beyer, L. Scaling Vision Transformers. Preprint at https://doi.org/10.48550/arXiv.2106.04560 (2022).

53. Gustafsdottir, S. M. *et al.* Multiplex Cytological Profiling Assay to Measure Diverse Cellular States. *PLOS ONE* **8**, e80999 (2013).

54. 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).

55. Rohban, M. H. *et al.* Systematic morphological profiling of human gene and allele function via Cell Painting. *eLife* **6**, e24060 (2017).

56. CLOOME: contrastive learning unlocks bioimaging databases for queries with chemical structures | bioRxiv. https://www.biorxiv.org/content/10.1101/2022.11.17.516915v2.

57. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).

58. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. Preprint at https://doi.org/10.48550/arXiv.1912.01703 (2019).

59. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).