Self-supervision advances morphological profiling by unlocking powerful image representations

Vladislav Kim[1§*], Nikolaos Adaloglou[1,2*], Marc Osterland[1], Flavio M. Morelli[1], Marah Halawa[1,3], Tim König[4], David Gnutt[4], 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

[3]Computer Vision and Remote Sensing, Technical University Berlin, Berlin, Germany

[4]Image-Based Screening Systems, Bayer AG, Wuppertal, Germany

[*]Equal contributions

[§]Corresponding author

Abstract

Cell Painting is an image-based assay that offers valuable insights into drug mechanisms of action and off-target effects. However, traditional feature extraction tools such as CellProfiler are computationally intensive and require frequent parameter adjustments. Inspired by recent advances in AI, we trained self-supervised learning (SSL) models DINO, MAE, and SimCLR on subsets of the JUMP-CP dataset to obtain powerful image representations for Cell Painting. We assessed the reproducibility and biological relevance of SSL features and uncovered the critical factors influencing model performance, such as training set composition and domain-specific normalization techniques. Our best model (DINO) surpassed CellProfiler in drug target and gene family classification, significantly reducing computational time and costs. All SSL models showed remarkable generalizability without fine-tuning, outperforming CellProfiler on an unseen dataset of genetic perturbations. Our study demonstrates the effectiveness of SSL methods for morphological profiling, suggesting promising research directions for improving the analysis of related image modalities.

Introduction

Morphological profiling uses image-based readouts to characterize the effect of chemical and genetic perturbations[1–4] based on alterations in 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 that utilizes 5 fluorescent dyes to stain 8 cellular compartments[13], generating thousands of morphological measurements per cell through automated image analysis. These high-dimensional readouts are used for hypothesis-free compound profiling, differentiating Cell Painting from target-based approaches. Despite significant progress in the visual representation learning field[14–17], the analysis of Cell Painting images still largely relies on classical computer vision techniques[18].

Conventional morphological profiling starts with single-cell segmentation, using 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 and feature selection methods are applied to reduce redundancy[21]. This multi-step workflow is computationally intensive and often requires adjustment of segmentation parameters when applied to new datasets. By contrast, deep learning models can offer a computationally efficient and segmentation-free alternative to morphological profiling.

The limited availability of biological labels has largely restricted the application of supervised learning in Cell Painting. Instead, morphological profiles are used for construction of biological maps[22,23] to identify phenotypes and modes of action using the guilt-by-association principle. Specifically, clustering compounds or genes by morphological similarity provides mechanistic insights from annotated cluster members. Alternatively, classifiers trained on extracted features can predict downstream tasks, such as drug toxicity[24] or cell health phenotypes[25]. However, label scarcity generally precludes end-to-end supervised learning from images, with only few exceptions[26]. The recently released JUMP-CP dataset[27] provides unprecedented opportunities for developing novel AI-based feature extraction methods. This large-scale image set (115 TB) contains approximately 117,000 chemical and 20,000 genetic perturbations. However, most compounds lack annotations, with only 4.5% having experimentally elucidated bioactivity. Thus, leveraging the full potential of this dataset requires techniques that do not rely on data curation or biological annotations.

Self-supervised learning (SSL) methods learn feature representations from unlabeled data through a pretext task. Early SSL pretext tasks focused on predicting image transformations[28,29]. However, the current state-of-the-art performance has been achieved through methods that maximize the agreement between transformed views of the same image. For instance, PIRL[30], MoCo[31] 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 was overcome by recent non-contrastive approaches such as BYOL[32] 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[16,33,34].

Recent advances in the SSL field have been accelerated by the adoption of the vision transformer[35] (ViT) architecture. ViTs operate on image patches projected into tokens and use self-attention[36] to capture global and local relationships between 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] and masked Siamese networks[37], which achieved state-of-the-art results on natural images. Notably, ViT performance scales favorably with data volume and complexity[38], making these models well-suited for the analysis of high-throughput imaging data.

Prior work has explored various approaches for single-cell feature extraction from high-content images, including transfer learning[39], image inpainting[40], variational autoencoders[41], supervised[42], self-supervised[23,43,44] and weakly supervised learning[45–47]. However, existing single-cell methods require segmentation, leading to complex multistep workflows. Other approaches extract representations from whole images [48–51], but rely on scarce labels or pretrained ImageNet weights, which restricts their application to 3-channel images or requires embedding concatenation for multichannel (> 3) applications. To date, only few approaches[52,53] learn directly from microscopy images without segmentation or manually curated annotations. Moreover, a systematic study evaluating the benefits of SSL methods over classical analysis workflows for high-content imaging data is currently missing.

Here, we present the first 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 and assessed their generalizability on independent datasets with chemical and genetic perturbations. Crucially, we examined the requirements for the successful application of SSL in the fluorescence microscopy domain including relevant image augmentations, model architecture, training set composition and feature postprocessing techniques. We assessed the performance gap between supervised and SSL models for compound bioactivity prediction, elucidating scenarios favoring each approach. Our results indicate that SSL methods provide a robust, efficient, and segmentation-free alternative to CellProfiler and that SSL features enable accurate prediction of compound properties comparable to supervised models.

Results

SSL framework for segmentation-free morphological profiling

Our self-supervised learning (SSL) framework operates directly on image crops without cell segmentation (see Methods). We adapted 3 state-of-the-art SSL approaches (**Fig. 1a**) for 5-channel Cell Painting images: SimCLR (simple framework for contrastive learning of visual representations)[15], DINO (distillation with no labels)[16] and MAE (masked autoencoder)[17]. We pretrained all SSL methods on two JUMP-CP[27] data subsets: single-source and multisource training sets (see Methods). Using these pretrained models, we extracted features to construct morphological profiles of chemical and genetic perturbations in held-out evaluation sets (**Fig. 1c**). We benchmarked the SSL features 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[35] (see Methods).

We used small (ViT-S) and base (ViT-B) vision transformer architectures as SSL backbones that encode images into feature vectors. At inference, we split images into equally-sized crops that were input into a pretrained ViT, with the image feature obtained by averaging the crop features (see Methods). To correct for plate and experimental batch effects, we tested several normalization methods and selected the best postprocessing strategy for each feature type (see Methods and **Extended Data Fig. 1-2**). We generated perturbation profiles by averaging normalized features across replicates of the same perturbation. As JUMP-CP datasets contain several data sources and experimental batches (see Methods), this aggregation was conducted at the batch, source, and full dataset levels (see Methods), enabling assessment of reproducibility across batches and sources. Full dataset aggregation produced consensus profiles, which were used for drug target and gene family classification.

Benchmarking feature extraction methods on JUMP-CP data

We evaluated all feature extraction methods on 3 held-out JUMP-CP[27] data subsets (**Fig. 1c**). The first two subsets contained target-annotated compounds with 2 drugs per target class (see

Methods). This allowed us to assess the suitability of pretrained features for few-shot learning, an important task in morphological profiling, where only few examples per class are available. The third evaluation set consisted of gene overexpression perturbations (see Methods) from a data source not used in training. Including genetic perturbations allowed us to test the models' ability to generalize to previously unseen perturbations, since our SSL models were trained only on images of chemically perturbed cells. As all JUMP-CP perturbations were screened across multiple experimental batches in different laboratories (see Methods), we could also assess batch and data source effects.

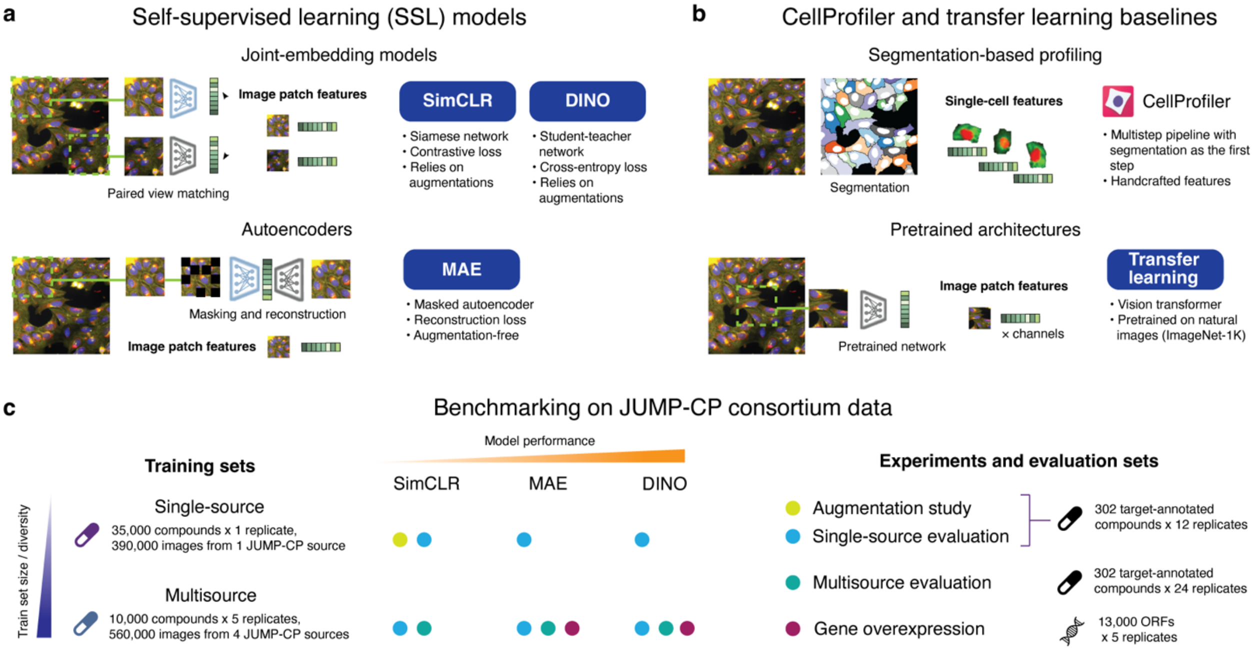

a
Self-supervised learning (SSL) models
Joint-embedding models
Image patch features
Paired view matching
SimCLR
• Siamese network
• Contrastive loss
• Relies on augmentations
DINO
• Student-teacher network
• Cross-entropy loss
• Relies on augmentations
Autoencoders
Masking and reconstruction
Image patch features
MAE
• Masked autoencoder
• Reconstruction loss
• Augmentation-free
b
CellProfiler and transfer learning baselines
Segmentation-based profiling
Segmentation
Single-cell features
CellProfiler
• Multistep pipeline with segmentation as the first step
• Handcrafted features
Pretrained architectures
Pretrained network
Image patch features
x channels
Transfer learning
• Vision transformer
• Pretrained on natural images (ImageNet-1K)
c
Benchmarking on JUMP-CP consortium data
Training sets
Train set size / diversity
Single-source
35,000 compounds x 1 replicate, 390,000 images from 1 JUMP-CP source
Multisource
10,000 compounds x 5 replicates, 560,000 images from 4 JUMP-CP sources
Model performance
SimCLR
MAE
DINO
Experiments and evaluation sets
Augmentation study
Single-source evaluation
302 target-annotated compounds x 12 replicates
Multisource evaluation
302 target-annotated compounds x 24 replicates
Gene overexpression
13,000 ORFs x 5 replicates

Figure 1: Self-supervised learning for morphological profiling.
a) Schematic of the SSL models used in this study. All models are segmentation-free and use only image crops as input. SimCLR and DINO were trained on the pretext task of matching features from augmented views of the same image. MAE was trained on the image reconstruction task with partially masked input. **b)** Schematic of the two baseline methods. CellProfiler: conventional method based on single-cell segmentation and handcrafted features. Transfer learning: pretrained vision transformer that outputs per-channel features. **c)** 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.

We compared feature extractors based on two key criteria: reproducibility and biological relevance (see Methods). To assess reproducibility, we used *perturbation mAP* (*mAP:* mean average precision), which quantifies the agreement between replicates of the same perturbation across experimental batches. Biological relevance was evaluated by the agreement between perturbations with the same biological annotation, using *target mAP* as the metric (see Methods). Additionally, we used nearest neighbor (NN) accuracy of matching across experimental batches (*not-same-batch* or *NSB accuracy*) and across experimental batches and distinct perturbations (*not-same-batch-or-perturbation* or *NSBP accuracy*). For genetic perturbations, we additionally evaluated the clustering quality with respect to gene family labels, using the *adjusted mutual information* (*AMI*) metric (see Methods).

Establishing augmentations for Cell Painting images

SimCLR and DINO rely on image augmentations to generate views of the same image during training (**Fig. 2a**). While the relative importance of individual augmentations was previously determined for RGB 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 to each fluorescent channel independently (see Methods).

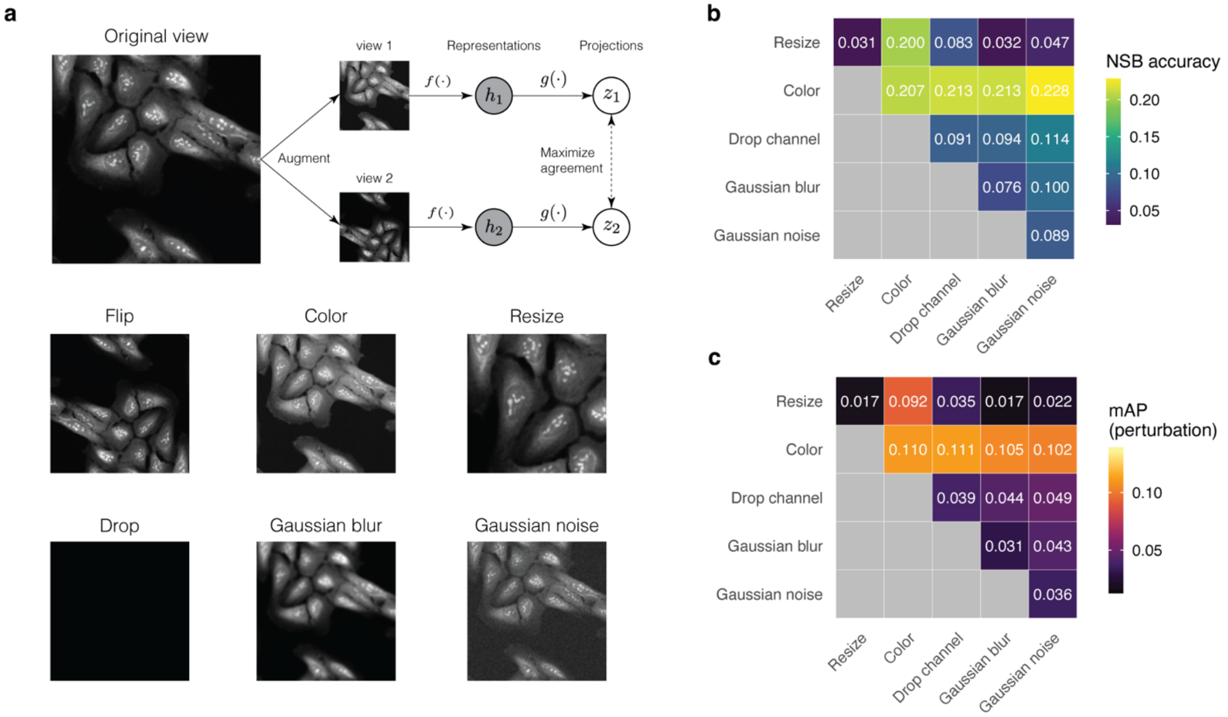


Figure 2: Augmentations for multichannel microscopy images.
a) Schematic of the SimCLR contrastive learning framework and representative images of the tested augmentations. Flip: vertical and horizontal flips. Color: per-channel stochastic intensity shift and brightness adjustment. Resize: random crop 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 15 SimCLR models with different combinations of augmentations trained and evaluated on the single-source data. 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*) (see Methods).

We pretrained 15 SimCLR models with different augmentation strategies (see Methods). A pairwise comparison of these models (**Fig. 2b-c**) revealed 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 in all subsequent experiments.

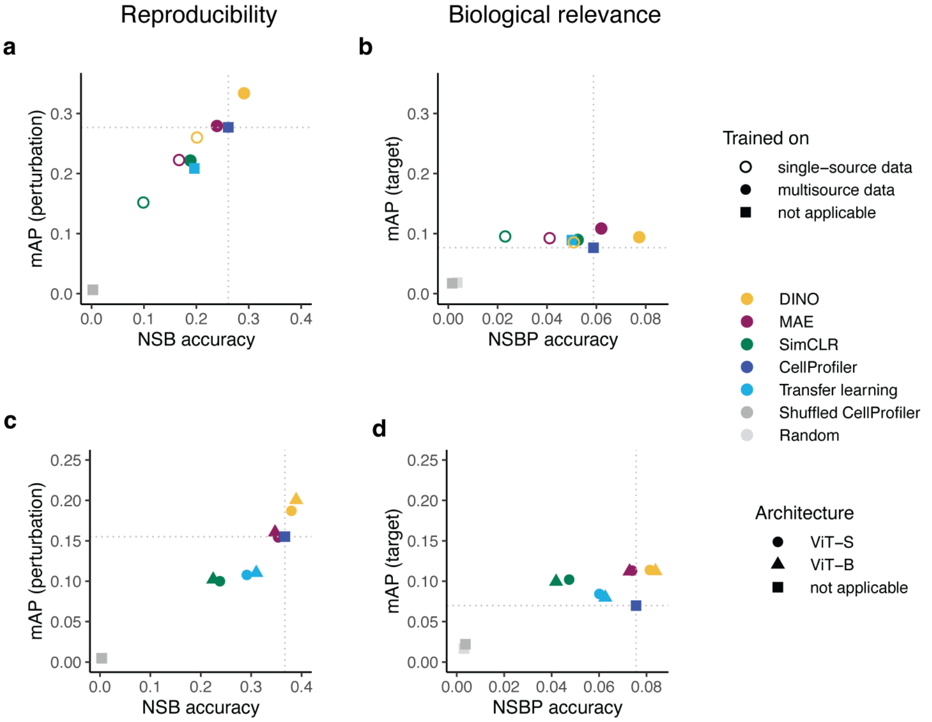


Figure 3: Impact of training data and model size on SSL model performance.
Performance comparison of SSL models (DINO, MAE, and SimCLR) and two baselines (CellProfiler and transfer learning) for different datasets (single-source and multisource) and model architectures (ViT-S and ViT-B). Left panels: 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). Dotted lines indicate CellProfiler performance. **a)-b)** Performance on the single-source evaluation set. Shapes indicate different training sets. Only the results for ViT-S architectures are reported. **c)-d)** Performance on the multisource evaluation set. Shapes indicate the ViT architecture. All SSL models were trained on the multisource training set.

DINO trained on multisource data outperforms CellProfiler

First, we evaluated the performance of DINO, MAE and SimCLR trained on different datasets (**Fig. 3a,b**). While all SSL models trained on single-source data performed worse than CellProfiler, models trained on multisource data match (MAE) or exceed (DINO) CellProfiler performance (**Fig. 3a,b**). DINO trained on multisource data achieved the best results among the SSL methods, surpassing CellProfiler. DINO features displayed better reproducibility (**Fig. 3a,c**) and biological relevance (**Fig. 3b,d**) compared to MAE and SimCLR. On the first evaluation set (**Fig. 3a-b**), DINO surpassed CellProfiler by a margin of 16% in *perturbation mAP*, 3% in *NSB accuracy,* 22% in *target mAP,* and 32% in *NSBP accuracy*. On the second evaluation set (**Fig. 3c-d**), DINO outperformed CellProfiler by even greater margins: 29% in *perturbation mAP*, 6%

in *NSB accuracy*, 61% in *target mAP*, and 11% in *NSBP accuracy*. Notably, transfer learning features showed the worst performance, encouraging the use of SSL methods for morphological profiling.

The superiority of DINO was even more evident in F1-score curves (**Extended Data Fig. 3**) and further confirmed by comparing reproducibility across JUMP-CP data sources (**Extended Data Fig. 4**, see Methods). DINO yielded similar or better performance to CellProfiler in 3 out of 4 JUMP-CP data sources (**Extended Data Fig. 4a-d**) and achieved higher *perturbation mAP* (+73%) and *NSB accuracy* (+2%) on a cross-source matching task (**Extended Data Fig. 4e**, see Methods).

Interestingly, models based on the larger ViT-B architecture only showed a marginal improvement over the ViT-S models (**Fig. 3c,d**). This was observed for both the SSL methods and the transfer learning baseline. Our results indicate that the biggest performance gain was achieved by expanding the training set to a more extensive and diverse image set as opposed to increasing the model size. In subsequent evaluations, we focused solely on SSL models trained on the multisource data using the ViT-S architecture, motivated by its lower compute requirements.

UMAP embeddings reveal biological and technical axes of variation

Next, we used UMAP[54] (see Methods) to embed DINO, MAE, SimCLR and CellProfiler features in 2 dimensions. To assess whether feature embeddings produced biologically meaningful clusters, we highlighted a selection of 20 drug targets in the UMAP space (see Methods). All embeddings grouped compounds with the same target to some extent (**Fig. 4**). Notably, DINO and CellProfiler embeddings yielded well-separated clusters in the UMAP, highlighting targets such as NAMPT, PAK1, and RET (**Fig. 4a, d**). Additionally, DINO embeddings demonstrated superior cluster separation for KRAS, AKT1, DNMT3A, TGFBR1, CDK2, and CDK7 (**Fig. 4a**) compared with CellProfiler (**Fig. 4d**). These results further support that DINO features are biologically meaningful and at least as powerful as those of CellProfiler.

We then examined feature robustness with respect to technical sources of variation by coloring UMAP embeddings by experimental batch and data source (**Extended Data Fig. 5a,b**). SSL feature embeddings were more susceptible to technical variations compared to CellProfiler, displaying a stronger separation between experimental subgroups. Upon closer inspection, we found that the most pronounced data source effects occurred within DMSO negative controls (**Extended Data Fig. 5a,c**) and that the differences strongly correlated with variations in cell count (**Extended Data Fig. 5d**). We hypothesize that CellProfiler is more robust towards cell count variations since the features are extracted from single cells. Additionally, we quantified the impact of technical variation (see Methods) and found that both SSL and CellProfiler features were affected by experimental batch and source effects to some extent (**Extended Data Fig. 6**). These results suggest that SSL representations capture both biological and technical axes of variation in the data.

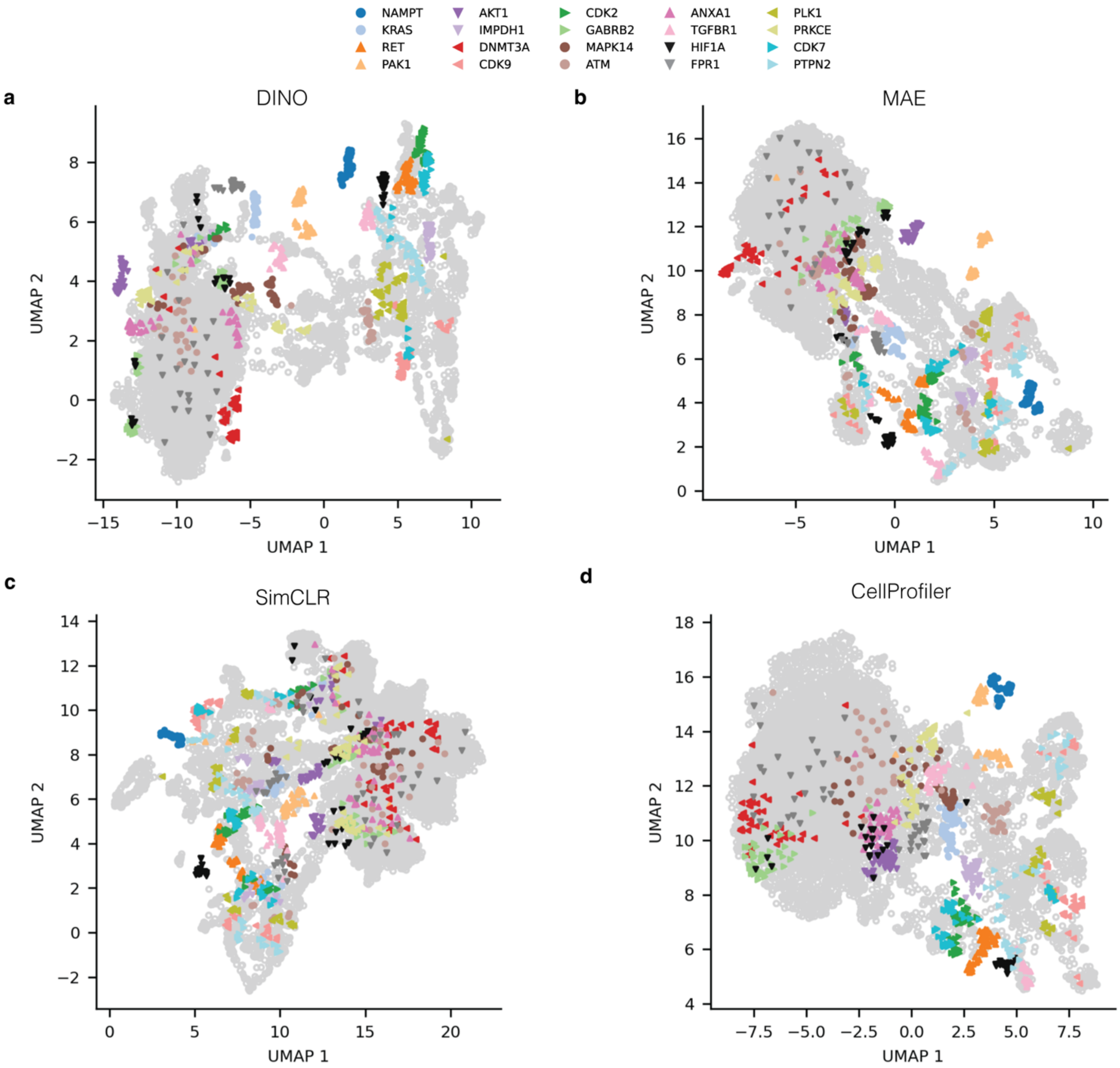


Figure 4: UMAP embeddings of SSL and CellProfiler features reveal compound clusters. Two-dimensional embeddings of well-level features from SSL methods and the CellProfiler baseline on the multisource evaluation set. Colors highlight target labels for a selection of 20 targets (see Methods). Perturbations with other targets are depicted as grey hollow points. All SSL models used the ViT-S architecture.

DINO features generalize to genetic perturbations and recapitulate gene families

To evaluate the generalizability of SSL models, we used an independent dataset of gene overexpression perturbations from a new source (see Methods). As for chemical perturbations, DINO features demonstrated the best reproducibility (**Fig. 5a**), with MAE achieving comparable performance. We also assessed the ability to predict gene family labels (see Methods) and found that all SSL features outperformed CellProfiler on metrics of biological relevance (**Fig. 5b-c**). The most notable improvements were seen with DINO and MAE features that improved gene family predictions by 41% in *NSBP accuracy* over CellProfiler.

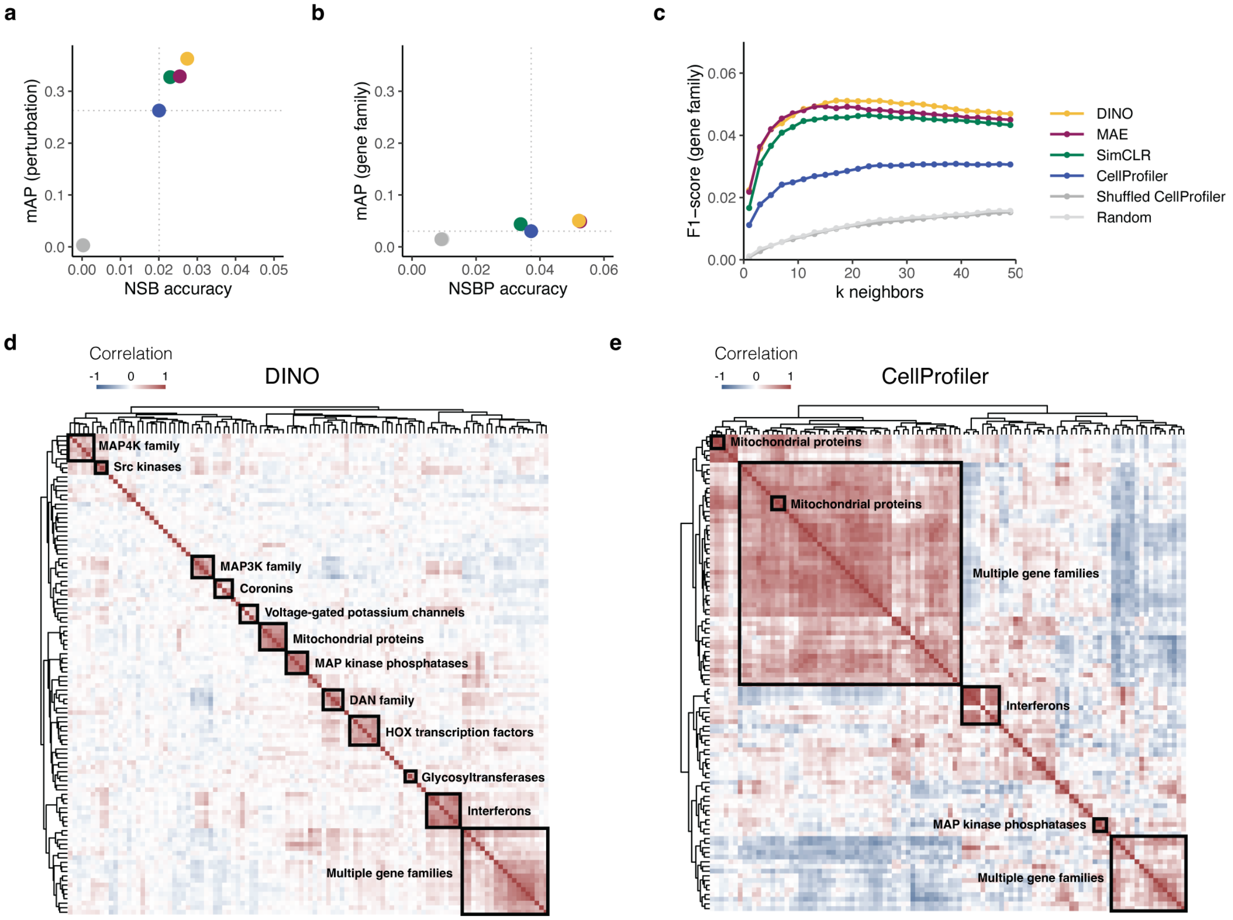


Figure 5: Evaluation of SSL models on genetic perturbations demonstrates generalizability to unseen data. Evaluation of SSL models and CellProfiler on an independent gene overexpression set. SSL models were trained only on images of chemically perturbed cells. Colors indicate different models and two randomized baselines: Shuffled CellProfiler (CellProfiler features with shuffled labels) and Random (random normally distributed features). Dotted lines in a)-b) indicate CellProfiler performance. **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 gene 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 **Extended Data Fig. 7-8**.

Using pairwise gene similarity analyses, we tested the ability of morphological features to recapitulate gene families. For a selection of 20 gene families (see Methods), we performed hierarchical clustering of gene profiles for DINO (**Fig. 5d**) and CellProfiler (**Fig. 5e**). The resulting similarity maps were annotated to highlight groups based on gene and gene family labels (see Methods). Our qualitative comparison (**Fig. 5d and Extended Data Fig. 7**) revealed that DINO features recovered a larger number of gene groups and produced more homogeneous clusters than CellProfiler. DINO recapitulated 11 gene groups, including MAP4K family, Src kinases, MAP3K family, coronins, voltage-gated potassium channels, mitochondrial proteins, MAP kinase phosphatases, DAN family, HOX transcription factors, glycosyltransferases, and interferons. By contrast, CellProfiler recovered only 3 gene groups (mitochondrial proteins, interferons, and MAP kinase phosphatases) and 2 large clusters of mixed gene families (**Fig. 5e and Extended Data Fig. 8**).

To provide a more objective assessment, we computed the adjusted mutual information (AMI) between cluster assignments and gene family labels (see Methods). We found that DINO (AMI = 0.51) outperformed CellProfiler (AMI = 0.19) on the gene clustering task, indicating that DINO features excel at capturing gene family information. Since all SSL models were trained on images with compound-treated cells, these results demonstrate the remarkable generalizability of SSL models, enabling their application to unseen data sources and conditions without parameter adjustments.

DINO enables bioactivity prediction comparable to supervised CNNs

In [26], convolutional neural networks (CNNs) were used to predict compound activity across 209 ChEMBL assays from Cell Painting images. To evaluate the performance gap between supervised and self-supervised learning, we compared bioactivity prediction models trained on DINO features versus images directly. Specifically, we assessed a neural network (NN) trained on DINO features (see Methods) against 6 CNNs trained on Cell Painting images from [26]. We used DINO pretrained on the JUMP-CP data, allowing us to probe its out-of-distribution generalizability. As an additional baseline, a NN trained on CellProfiler features was incorporated from [26].

After ranking all methods by mean AUCROC across 209 assays (see Methods), we found (**Extended Data Table 1**) that the model trained on DINO features (AUCROC = 0.72) achieved performance comparable to GapNet (AUCROC = 0.73), the third best method. Although the top 3 CNNs had slightly higher mean AUCROC values, DINO outperformed 3 additional CNNs and a model trained on CellProfiler features (**Extended Data Table 1**). Among the 8 methods compared, DINO ranked 4th for the number of assays predicted with AUCROC above 0.7 and 0.8. This confirms that DINO can generalize to novel datasets and tasks without fine-tuning.

Nevertheless, the top 3 CNNs surpassed the DINO-based model in the number of assays predicted with AUCROC > 0.9, indicating a performance gap. Further analysis revealed that the model trained on DINO features offered comparable or better performance on assays with limited data (**Extended Data Fig. 9**). For assays with only few (< 100) activity labels available, the DINO model produced a similar number of accurate predictions as the top 3 CNNs (**Extended Data Fig. 9a**). However, on assays with 100-500 labels, the CNNs showed better performance than the DINO model (**Extended Data Fig. 9a**). When considering only assay predictions with AUCROC > 0.7 (**Extended Data Fig. 9b**), the DINO model achieved higher median AUCROC values for assays with few activity labels (< 50 and 50-100), consistent with SSL features excelling in few-shot settings[55].

SSL pipeline is significantly faster than CellProfiler

As efficient data processing is crucial for accelerating research throughput, we additionally benchmarked the computation time and cloud costs of feature extraction using DINO versus CellProfiler. For this comparison, we used 12 GPU-accelerated cloud instances for DINO and 12 CPU-intensive instances for CellProfiler (**Extended Data Table 2**). We found that DINO was 50 times faster than CellProfiler, with an average processing time of 1.3 minutes per plate

(**Extended Data Table 2**). Despite the need for GPU resources, the cloud costs per plate were over 50 times lower for DINO than for CellProfiler (**Extended Data Table 2**). Moreover, DINO offers a simpler workflow which processes images end-to-end, in contrast to the multi-step CellProfiler workflow which requires illumination correction, segmentation, feature extraction and selection.

Taken together, our findings show that image-level SSL methods are a viable alternative to traditional segmentation-based approaches, offering improved performance, generalizability to new datasets, speed, and lower workflow complexity and computational costs.

Discussion

To assess the applicability of SSL for Cell Painting, we trained and evaluated 3 state-of-the-art methods DINO, MAE, and SimCLR on complex datasets with chemical and genetic perturbations. Using reproducibility and biological relevance as our main criteria, we showed that our best model, DINO, outperformed the established feature extraction tool, CellProfiler, in drug target and gene family classification, with even greater improvements in gene clustering.

Our SSL models captured informative cell-related features that generalized to unseen datasets without parameter fine-tuning. While trained only on compound perturbations, DINO achieved superior classification and clustering performance on a novel gene overexpression set, facilitating the construction of biological maps[22]. For compound activity prediction, DINO features transferred remarkably well to a new dataset, with the model trained on DINO features achieving comparable performance to CNNs[26] trained on that dataset directly. We hypothesize that the strong transferability[56] of DINO can be attributed to its image-level pretext task that effectively captures low-frequency signals[57] associated with cellular shape. The generalizability of our SSL models expedites the analysis of new datasets in contrast to CellProfiler, which requires frequent parameter adjustments.

Previous studies[45,47,51] fine-tuned ImageNet-pretrained networks to learn representations for Cell Painting, often curating the training set through compound preselection. These methods process each channel independently and output concatenated channel representations, increasing computational complexity and feature redundancy. By contrast, our SSL models are tailored for uncurated 5-channel images, resulting in compact representations with lower redundancy. More recently, DINO was applied for learning single-cell morphological representations[43,44], with [44] reporting superior performance over CellProfiler. However, unlike these approaches, our SSL framework operates without cell segmentation, streamlining feature extraction.

Our study offers practical guidance for applying SSL methods to microscopy images. We systematically examined the role of augmentations, confirming the importance of 'Color' augmentation observed for natural images[15]. We found, however, that the 'Resize' augmentation decreased representation quality. We hypothesize that enforcing scale-invariance is detrimental, as cell size variations in microscopy images reflect actual phenotypic changes. Additionally, training set size and heterogeneity played a crucial role in SSL model performance. Scaling from a single-source to a multisource training set improved performance more than using larger vision transformer architectures, which may require more training data and longer pretraining[58]. Notably, masked image modelling (MAE) performed comparably to self-distillation pretraining (DINO), which suggests that combining both SSL pretext tasks could further improve representations as shown for natural images[59,60].

Our analysis (**Extended Data Fig. 9**) revealed that models trained on SSL features excel in bioactivity prediction when ground-truth labels are scarce, while dedicated supervised methods achieve superior performance given ample labeled data. Since compound annotations are sparse, training supervised models directly from images remains infeasible in most but few[26] cases, which makes the use of SSL features an attractive alternative to harness the large-scale unlabeled data such as the JUMP-CP dataset.

With a 50-fold reduction in compute time and costs compared to CellProfiler, SSL feature extraction methods can facilitate compound screening campaigns of unprecedented scale, revolutionizing the pace of early drug discovery. However, pretraining a DINO model demands substantial compute resources, requiring approximately 300 GPU hours. Given the intensive resource requirements, our study used only subsets of the JUMP-CP dataset, leaving room for exploration of the full dataset's potential. Investigating the emerging properties of larger self-supervised ViTs trained on the complete JUMP-CP dataset offers promising research directions.

One limitation of our segmentation-free approach is that it operates at image-crop level and does not provide insights into cell heterogeneity, making CellProfiler a more suitable tool for single-cell analyses. Additionally, CellProfiler provides interpretability, by linking individual features to specific microscopy channels and mathematically defined morphological descriptors. However, even with self-supervised ViTs, we can gain some level of interpretability by examining self-attention maps (see **Extended Data Fig. 10**). Furthermore, SSL methods showed higher susceptibility to experimental batch and laboratory effects compared to CellProfiler. Post-hoc approaches like Harmony[61] can mitigate the effect of technical variation on SSL features. Alternatively, incorporating batch alignment as an additional objective during pretraining may produce more robust SSL representations.

Our SSL models showed generalizability across Cell Painting datasets but remain limited in transferability across other microscopy modalities, requiring fine-tuning for assays with different staining than Cell Painting. Drawing inspiration from natural language and image domains[62–65], we encourage the development of assay-agnostic foundation models for microscopy images, which can standardize and expedite the analysis of high-content assays across various imaging modalities.

Methods

Technical terminology

Image *features* are high-dimensional readouts extracted from images using segmentation-based or deep learning approaches. We use the terms “*representations*” and “*features*” interchangeably. Feature vectors can be embedded into 2 dimensions for visualization; we refer to these projections as *embeddings*.

Cell Painting assay is conducted in 384-well *plates*, with each well imaged at several locations to produce multiple images or fields of view (FOVs). *Well profiles* or *well features* refer to features aggregated for each well across multiple FOVs. Cell Painting screening is performed in experimental *batches* containing groups of plates. Unless specified otherwise, the term “batch” refers to an experimental batch and not to a minibatch used for training deep learning models. In the JUMP Cell Painting consortium, several *laboratories* or *data sources* generated the data, adding a hierarchical level above the *plate* and *batch* levels.

Cells in each well are treated with a specific *perturbation* (e.g., compound or gene overexpression). This provides *perturbation labels* to assess reproducibility across repeated measurements or *replicates*. We used several subsets of the JUMP-CP data, which we refer to as *datasets*. Aggregating perturbation features across all replicates in a dataset produces a *consensus profile*. For a small subset of compounds, we have drug *target labels*, i.e., proteins targeted by these drugs. Gene overexpression perturbations correspond to individual genes which can be annotated and grouped in *gene families*. To evaluate biological relevance, we used *target labels* for compounds and *gene family labels* for genetic perturbations.

JUMP-CP training and validation sets

We used subsets of the JUMP Cell Painting dataset[27] (cpg0016-jump) for training and evaluation of self-supervised learning (SSL) models. The complete JUMP-CP dataset (115 TB) includes 116,750 chemical perturbations, 12,602 gene overexpression and 7,975 CRISPR perturbations probed in a human cancer cell line (U2OS) in 5 replicates. Each chemical perturbation was screened by 5 out of the 10 consortium laboratories ("sources") that used a standardized protocol but possibly different instrumentation. Genetic perturbations were screened solely by sources 4 and 7.

For model training, we used only images of cells treated with chemical perturbations. We used two training sets: a single-source and a multisource training set. 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. An overview of the JUMP-CP batches and plates used for model training is provided in **Supplementary Data 1**.

To assess our models, we used JUMP Target2 plates[27] that contain 306 compounds with drug target labels. These plates were imaged in every experimental batch, enabling us to not only assess model performance using biological labels, but also evaluate batch and laboratory effects. Single-source (6 batches, 33,962 images) and multisource (16 batches, 75,545 images) validation sets were constructed using JUMP-Target2 plates from the respective sources of the single-source and multisource training sets. JUMP-CP batches and plates used for evaluation of models are listed in **Supplementary Data 2**.

JUMP-CP gene overexpression test set

The JUMP-CP[27] gene overexpression data (source 4) with Open Reading Frames (ORFs) was used for the final assessment of 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 5,198 ORFs. Gene group memberships were assigned to each ORF perturbation using the HUGO Gene Nomenclature Committee (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. 1,970 ORF perturbations satisfied this criterion. The complete list of batches and plates is provided in **Supplementary Data 2**.

Image preprocessing

The JUMP-CP consortium[27] generated Cell Painting images with 5 color 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 trained 15 SimCLR models with the ResNet-50 backbone to test different image augmentation strategies. All models were trained and evaluated on the single-source data. We probed pairwise combinations of 5 augmentations: 'Resize', 'Color', 'Drop channel', 'Gaussian noise' and 'Gaussian blur'. The 'Flip' augmentation that rotates an image by 180 degrees along the horizontal or vertical axes was used by default. '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: $I_{c,i,j} = I_{c,i,j} + \varepsilon,\ \varepsilon \in U(-0.3, 0.3)$ and a random brightness change: $I_{c,i,j} = I^{\gamma}_{c,i,j},\ \gamma \in U(0.5, 1.5)$ 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: $I_{c,i,j} = I_{c,i,j} + \mu_{i,j},\ \mu_{i,j} \sim N(0, 0.05)$. 'Gaussian blur' applies a Gaussian filter with a kernel size of 23 pixels 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 pixels) from the images and provided these as inputs to the models. 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. For SimCLR and DINO, we additionally applied the augmentation pipeline, 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 pixels. 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 1**.

DINO was trained with a minibatch 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 minibatch 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 minibatch 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

A feature extraction model maps an input image $I \in R^{C \times H \times W}$ to a *d*-dimensional feature space through a mapping function $f: R^{C \times H \times W} \rightarrow R^{d}$. In DINO, MAE, and SimCLR, the mapping $f(\cdot)$ is performed by a vision transformer (ViT) backbone. The dimensionality d of learned features depends on the network architecture, with $d = 384$ for ViT-S and $d = 768$ for ViT-B.

At inference, each microscopy image I, corresponding to a single field of view (FOV), was split into 224x224 image crops $\{x_i, i = 1, \dots, N_{crops}\}$. All image crops were passed through a pretrained ViT backbone to generate crop features $f(x_i)$. Crops with no cells were excluded following the same criteria used during training. The resulting image crop features were aggregated using the arithmetic mean: $f_{\text{img}} = \frac{1}{N_{\text{crops}}} \sum_{i=1}^{N_{\text{crops}}} f(x_i)$. Well features were obtained by taking the mean across all FOV images: $f_{\text{well}} = \frac{1}{N_{\text{FOV}}} \sum_{k=1}^{N_{\text{FOV}}} f_{\text{img}}^{k}$.

We tested several feature postprocessing methods (**Extended Data Fig. 1-2**) and chose “sphering + MAD robustize” for SSL features. First, we removed well features with variance less than $1 \cdot 10^{-5}$. We then applied a sphering transformation[39] Φ to the remaining well features, followed by normalization using whole-plate median (med) and median absolute deviation (MAD):

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

To generate perturbation profiles for downstream analyses, we averaged normalized well features f_{norm} across multiple replicates. Aggregation was performed at several levels: *batch-aggregated profiles* average all replicates within an experimental batch, *source-aggregated profiles* average all replicates within a JUMP-CP data source, and *consensus profiles* average all replicates within an entire dataset (single-source/multisource/gene overexpression).

CellProfiler features

We used CellProfiler features provided by the JUMP-CP consortium (https://registry.opendata.aws/cellpainting-gallery/, for details see [27]). CellProfiler features were normalized using whole-plate median and MAD (“MAD robustize” normalization), which was the best postprocessing method for CellProfiler features (**Extended Data Fig. 1e**). We tested several feature selection approaches (**Extended Data Fig. 2**) and selected the set of 560 features from the CPJUMP1 study[66], 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[35] (ViT-S/16 or ViT-B/16) pretrained on the image classification task on ImageNet-1K was used to extract ‘transfer learning’ features. Each of the 5 channels was duplicated 3 times to generate pseudo-RGB images, which were individually passed through a pretrained ViT. The transfer learning features were obtained by concatenating individual channel features, resulting in $5 \cdot 384 = 1920$-dimensional feature vectors. As for SSL methods, low variance features ($< \ 1 \cdot 10^{-5}$) were removed before normalization. The transfer learning features were normalized using whole-plate median and MAD (“MAD robustize” normalization), which was the best postprocessing method for transfer learning (**Extended Data Fig. 1d**).

Evaluation of reproducibility and biological relevance

We evaluated all features based on two key criteria: *reproducibility* using perturbation labels and *biological relevance* using drug target or gene family labels (see “Technical terminology”). To assess the sensitivity and precision of inferring ground-truth labels based on pairwise feature distances $D\big(f_i, f_j\big)$, we followed an approach similar to [66]. For each perturbation i, we define a neighborhood $N_{i,d} = \{j \mid D\big(f_i, f_j\big) \leq \ d\}$ consisting of all other perturbations j within a cosine distance threshold d of perturbation i. We then compared the label y_i of perturbation i with the labels y_j of its nearest neighbors $\{ j \in N_{i,d}\}$. The precision $P_{i,d}$ and recall $R_{i,d}$ of matching labels for perturbation i at distance threshold d were calculated as:

$$P_{i,d} = \frac{\sum_{j \in N_d} I(y_i = y_j)}{|N_{i,d}|}$$

$$R_{i,d} = \frac{\sum_{j \in N_d} I(y_i = y_j)}{\sum_{j \neq i} I(y_i = \ y_j)}$$

where I is an indicator function, and $|N_{i,d}|$ is the size of the neighborhood of perturbation i.

Average precision (AP_i) was computed for each perturbation by varying the distance threshold d of the neighborhood $N_{i,d}$:

$$AP_i = \sum_{d} (R_{i,d} - R_{i,d-\Delta d}) P_{i,d}$$

Mean average precision (mAP) was then calculated by averaging the AP values across all perturbations:

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i$$

Perturbation mAP, measuring reproducibility, was estimated on batch-aggregated profiles (see “SSL inference and postprocessing”), thus quantifying the ability to match perturbations across batches. *Target mAP* was estimated on consensus profiles (see “SSL inference and postprocessing”) using drug target labels, focusing on biological content after technical variations were averaged out. For genetic perturbations, biological relevance was estimated using *gene family mAP*, calculated on consensus profiles with gene family labels. To evaluate the cross-source matching ability of features (**Extended Data Fig. 4e**), *perturbation mAP* was calculated on source-aggregated profiles. Along with AP values, F1-scores at *k* nearest neighbors were computed and visualized (**Extended Data Fig. 3**) to investigate whether some features worked better in specific *k* ranges.

The second class of metrics, widely used in morphological profiling[39,53,67], reports the nearest neighbor (NN) accuracy estimated on well profiles with restrictions on the possible match. To evaluate reproducibility, the *not-same-batch (NSB) accuracy* restricts true positive matches to well profiles from different experimental batches. The *not-same-batch-or-perturbation (NSBP) accuracy* restricts true positive matches to profiles from both different batches and distinct perturbations. We used *perturbation NSB accuracy* to evaluate feature reproducibility across batches and *target NSBP accuracy* to evaluate biological relevance.

UMAP embeddings

To visualize features in 2 dimensions, we generated UMAP (Uniform Manifold Approximation and Projection)[54] embeddings using the first 200 principal components as input, with the correlation distance as the metric. For 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 selected the 20 drug targets with the highest mean F1-scores from the JUMP-Target2 plate[27] annotation set. The F1-scores were determined by performing target classification using CellProfiler and DINO features. **Supplementary Table 2** provides the list of these 20 targets and their F1-scores stratified by feature type.

Quantification of technical biases

The impact of technical variation was assessed by examining well profiles of the multisource validation dataset, which contained 24 replicates of each perturbation (see “Technical terminology"). To quantify batch and source effects for each feature type, we compared within- and between-cluster similarity and connectivity, using batch and source information as cluster labels. We used 3 metrics: *Silhouette scores*[68], *Graph Connectivity (GC)*[69] and *Local Inverse Simpson's Index (LISI)*[69].

The silhouette score measures the similarity of an observation i to its own cluster (batch/source) relative to the nearest cluster[68]. It calculates the relative difference between the mean intra-cluster distance $a(i)$ and the mean nearest-cluster distance $b(i)$:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

The silhouette score ranges from -1 to 1, with higher values indicating the observation is well matched to its own cluster and poorly matched to neighboring clusters. We compared distributions of silhouette scores for all well profiles clustered by batch or source (**Extended Data Fig. 6c,f**).

The *GC* and *LISI* metrics are based on a *k*-nearest neighbor (kNN) graph *G(V, E)*. This graph consists of vertices *V* corresponding to well profiles. Each vertex is connected to its *k* nearest neighbors based on pairwise cosine distances defining the edge set *E*. Let C be a set of clusters, such as batches or sources. Taking only the vertices of a specific cluster $c \in C$ induces a subgraph $G_c(V_c, E_c)$. GC measures the ratio between the number of vertices in the largest connected component *(LCC*) of G_c and the total number of vertices in G_c, averaged across all clusters:

$$GC = \frac{1}{|C|} \sum_{c \in C} \frac{|LCC(G_c(V_c, E_c))|}{|V_c|}.$$

If the LCC of G_c is almost as large as G_c itself, this indicates that vertices from the same cluster are close together – a sign of batch/source effects. We reported GC for *k* = 1, 2, 3, 5, 10, 15 (**Extended Data Fig. 6a,d**).

LISI quantifies neighborhood diversity in the kNN-graph *G* using the inverse Simpson's index:

$$LISI = \frac{1}{\sum_{c \in C} p(c)^2},$$

where $p(c)$ is the relative abundance of cluster c. LISI can be interpreted as the expected number of profiles to be sampled before two are drawn from the same cluster[61]. Higher *LISI* implies more diverse neighborhoods and lower batch/source effects. LISI was calculated for *k* = 15, 30, 60, 90 (**Extended Data Fig. 6b,e**). For both *GC* and *LISI*, the values for *k* are based on [69].

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 features; 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 between 4 and 10 unique ORF perturbations. Since many of the gene families were heterogeneous and uncorrelated, we only selected the top 20 gene families with the highest within-family correlations in the respective feature space, resulting in 2 gene sets for DINO and CellProfiler (**Supplementary Data 3**).

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, adjacent 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:

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

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

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

Assessing the performance gap between supervised and self-supervised learning

We used DINO pretrained on the multisource JUMP-CP dataset to extract morphological features from Cell Painting images from a dataset of 30,000 small-molecule perturbations[70]. For bioactivity prediction, we only used 10,000 compounds with activity labels from a study[26], in which convolutional neural networks (CNNs) were trained to predict compound activity across 209 ChEMBL assays. We trained a 3-layer fully connected neural network (FNN) on the extracted DINO features to predict compound activity. To ensure comparability with the CNNs trained on Cell Painting images directly, we used the same code base (https://github.com/ml-jku/hti-cnn), activity labels and train/validation/test splits as [26].

We included 6 CNNs from [26] as fully supervised baselines for bioactivity prediction: GapNet, ResNet, DenseNet, MIL-Net, M-CNN and SC-CNN. All CNNs, except Single-Cell CNN (SC-CNN), were trained end-to-end on Cell Painting images without segmentation. A 3-layer FNN trained on CellProfiler features was incorporated as an additional baseline from [26]. The results for the CNNs and CellProfiler FNN were taken from the original publication. For details on the CNN and FNN architectures and training methodology, refer to [26].

Following [26], we evaluated the performance of the FNN trained on DINO features using area under the receiver operating characteristic curve (AUCROC) as the primary metric. The AUCROC values for each assay were obtained by averaging across 3 test splits. The mean AUCROC across 209 assays and the standard deviation are reported in **Extended Data Table**

1. Similarly, F1-scores were computed and the mean and standard deviation across all assays are also provided. To illustrate the performance gap across different data availability regimes, we grouped the assays into 5 bins based on the number of activity labels: [< 50, 50-100, 100-500, 500-1000, 1000-3000]. AUCROC values were visualized for these 5 assay groups in **Extended Data Figure 9**.

Implementation details

All self-supervised learning (SSL) models were implemented in Python 3.9.7 using PyTorch[71] v1.10.2 and PyTorch Lightning v1.6.3. SSL model training and inference were conducted on NVIDIA Tesla V100 GPUs with a VRAM of 32GB. Feature postprocessing (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[72] 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[61] 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 was 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 is provided in **Supplementary Code** and will be made publicly available on GitHub upon publication. The code is distributed under the BSD 3-Clause License. The model weights are provided and intended for non-commercial use only.

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 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 support.

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. MH trained a bioactivity prediction model on DINO features and compared it with supervised CNNs. VK designed the figures, with contributions from PAMZ, MO, FM and NA. VK, NA and PAMZ wrote the manuscript with inputs from MO, FM, TK and DG. All authors reviewed and approved the final version.

Competing interests

VK, MO, FM, MH, TK, DG, and PAMZ are employees of Bayer AG.

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. Celik, S. *et al.* Biological Cartography: Building and Benchmarking Representations of Life. 2022.12.09.519400 Preprint at https://doi.org/10.1101/2022.12.09.519400 (2022).

23. Sivanandan, S. *et al.* A Pooled Cell Painting CRISPR Screening Platform Enables de novo Inference of Gene Function by Self-supervised Deep Learning. 2023.08.13.553051 Preprint at https://doi.org/10.1101/2023.08.13.553051 (2023).

24. Garcia de Lomana, M., Marin Zapata, P. A. & Montanari, F. Predicting the Mitochondrial Toxicity of Small Molecules: Insights from Mechanistic Assays and Cell Painting Data. *Chem. Res. Toxicol.* **36**, 1107–1120 (2023).

25. Way, G. P. *et al.* Predicting cell health phenotypes using image-based morphology profiling. *Mol. Biol. Cell* **32**, 995–1005 (2021).

26. Hofmarcher, M., Rumetshofer, E., Clevert, D.-A., Hochreiter, S. & Klambauer, G. Accurate Prediction of Biological Assays with High-Throughput Microscopy Images and Convolutional Networks. *J. Chem. Inf. Model.* **59**, 1163–1171 (2019).

27. 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).

28. Gidaris, S., Singh, P. & Komodakis, N. Unsupervised Representation Learning by Predicting Image Rotations. Preprint at https://doi.org/10.48550/arXiv.1803.07728 (2018).

29. 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.

30. Misra, I. & Maaten, L. van der. Self-Supervised Learning of Pretext-Invariant Representations. in 6707–6717 (2020).

31. 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).

32. 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).

33. 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).

34. 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.

35. 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).

36. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems* vol. 30 (Curran Associates, Inc., 2017).

37. 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.

38. Dehghani, M. *et al.* Scaling Vision Transformers to 22 Billion Parameters. Preprint at http://arxiv.org/abs/2302.05442 (2023).

39. 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).

40. 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).

41. 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).

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

43. Pfaendler, R., Hanimann, J., Lee, S. & Snijder, B. Self-supervised vision transformers accurately decode cellular state heterogeneity. 2023.01.16.524226 Preprint at https://doi.org/10.1101/2023.01.16.524226 (2023).

44. Doron, M. *et al.* Unbiased single-cell morphology with self-supervised vision transformers. 2023.06.16.545359 Preprint at https://doi.org/10.1101/2023.06.16.545359 (2023).

45. 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).

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

47. 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).

48. 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).

49. 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).

50. 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).

51. 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).
52. 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).
53. 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.
54. 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).
55. Chen, T., Kornblith, S., Swersky, K., Norouzi, M. & Hinton, G. Big Self-Supervised Models are Strong Semi-Supervised Learners. Preprint at https://doi.org/10.48550/arXiv.2006.10029 (2020).
56. Ericsson, L., Gouk, H. & Hospedales, T. M. How Well Do Self-Supervised Models Transfer? Preprint at https://doi.org/10.48550/arXiv.2011.13377 (2021).
57. Park, N., Kim, W., Heo, B., Kim, T. & Yun, S. What Do Self-Supervised Vision Transformers Learn? Preprint at https://doi.org/10.48550/arXiv.2305.00729 (2023).
58. Zhai, X., Kolesnikov, A., Houlsby, N. & Beyer, L. Scaling Vision Transformers. Preprint at https://doi.org/10.48550/arXiv.2106.04560 (2022).
59. Zhou, J. *et al.* iBOT: Image BERT Pre-Training with Online Tokenizer. Preprint at https://doi.org/10.48550/arXiv.2111.07832 (2022).
60. Oquab, M. *et al.* DINOv2: Learning Robust Visual Features without Supervision. Preprint at https://doi.org/10.48550/arXiv.2304.07193 (2023).
61. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).

62. Brown, T. B. *et al.* Language Models are Few-Shot Learners. Preprint at https://doi.org/10.48550/arXiv.2005.14165 (2020).

63. Radford, A. *et al.* Learning Transferable Visual Models From Natural Language Supervision. Preprint at https://doi.org/10.48550/arXiv.2103.00020 (2021).

64. Yuan, L. *et al.* Florence: A New Foundation Model for Computer Vision. Preprint at https://doi.org/10.48550/arXiv.2111.11432 (2021).

65. Wu, C. *et al.* Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. Preprint at https://doi.org/10.48550/arXiv.2303.04671 (2023).

66. 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).

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

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

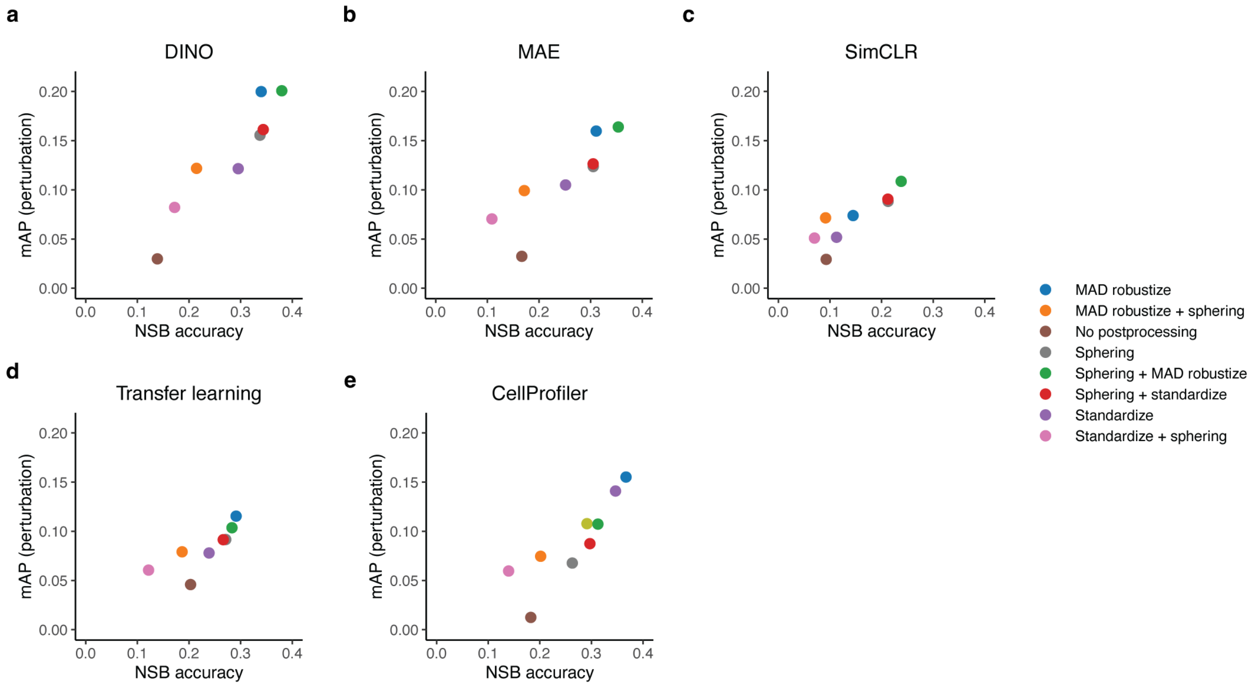
69. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).

70. 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).

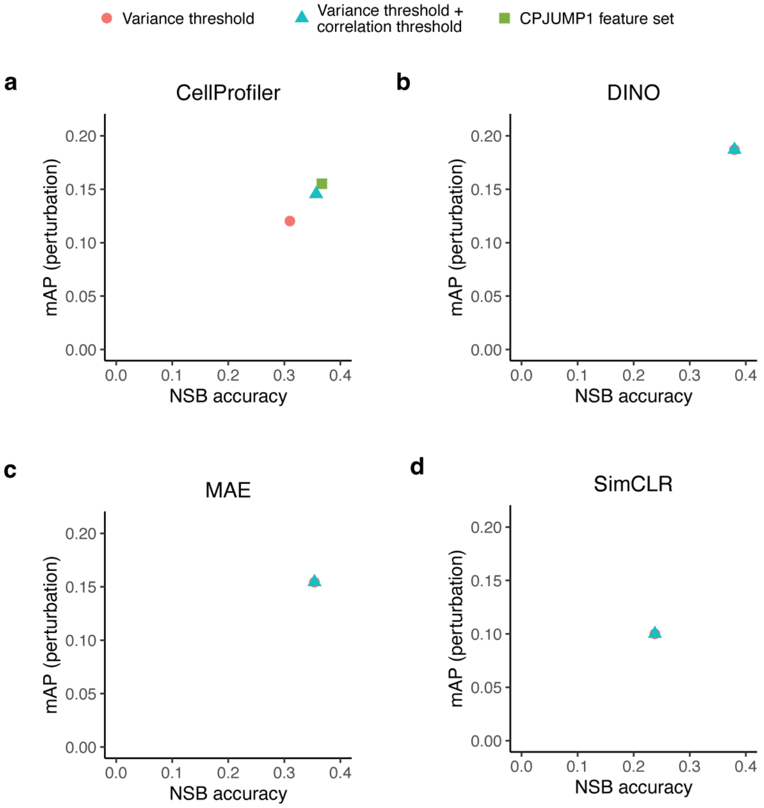
71. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. Preprint at https://doi.org/10.48550/arXiv.1912.01703 (2019).

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

Extended Data Figures

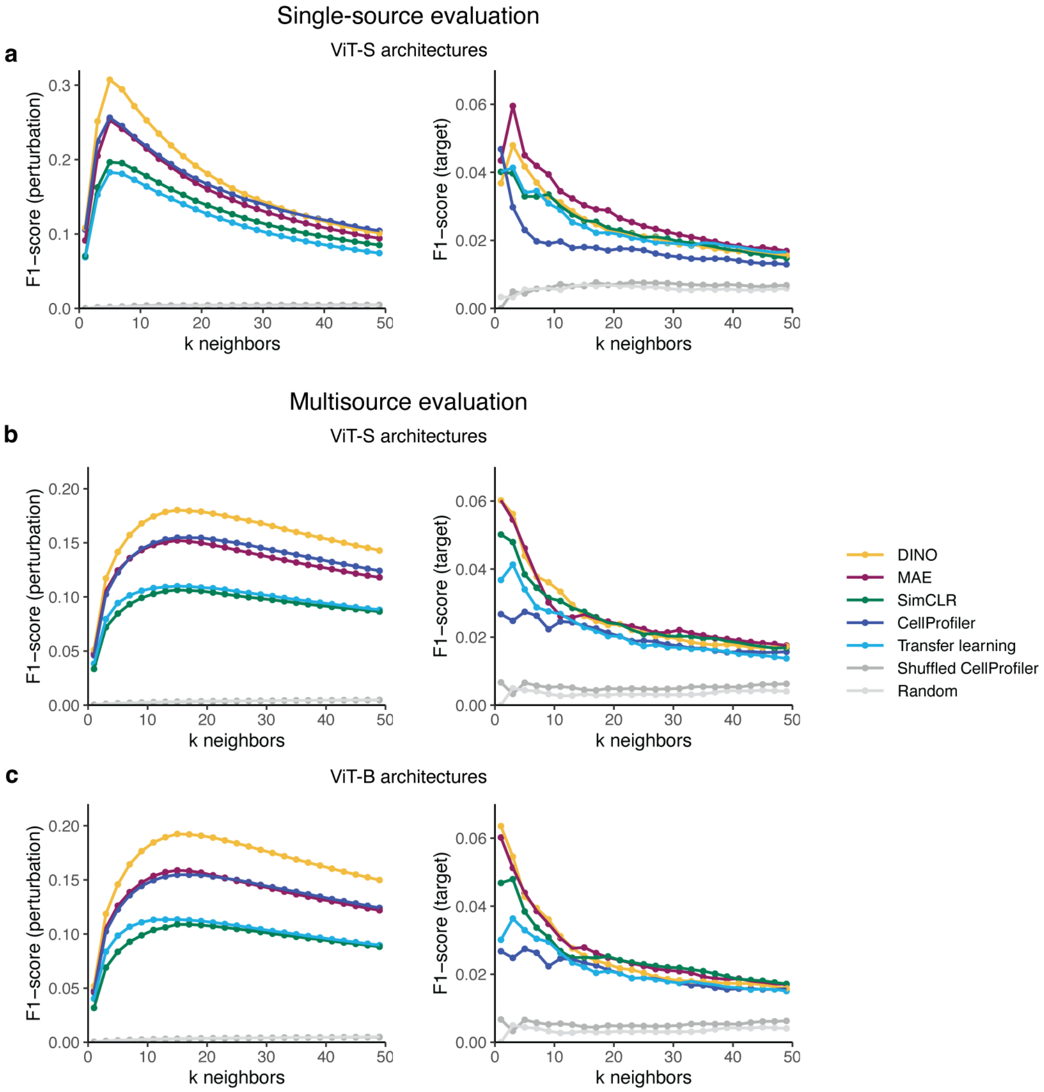


Extended Data Figure 1: Comparison of postprocessing methods for various representations.
Impact of normalization methods on reproducibility metrics *not-same-batch (NSB) accuracy* and *perturbation mean average precision (mAP)* computed on the multisource evaluation set. SSL models were trained on the multisource training set with the ViT-S architecture. Pairwise combinations of 3 normalization methods were tested. MAD robustize: center and scale each feature using plate median and median absolute deviation (MAD). Standardize: center and scale each feature using DMSO negative control mean and standard deviation. Sphering: center and scale using DMSO negative control mean and standard deviation and transform the data using the eigenvector matrix of the DMSO covariance matrix.



Extended Data Figure 2: Comparison of feature selection methods for various representations. Impact of feature selection methods on reproducibility metrics *not-same-batch (NSB) accuracy* and *perturbation mean average precision (mAP)* computed on the multisource evaluation set. SSL models were trained on the multisource training set with the ViT-S architecture. Two normalization methods were tested: Variance threshold was used to remove low-variance features, and variance threshold + correlation threshold was used to further eliminate redundant features. For CellProfiler, the CPJUMP1 feature set was also included, selected based on the CPJUMP1 dataset[1], which comprises chemical, ORF, and CRISPR perturbations.

[1] 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).

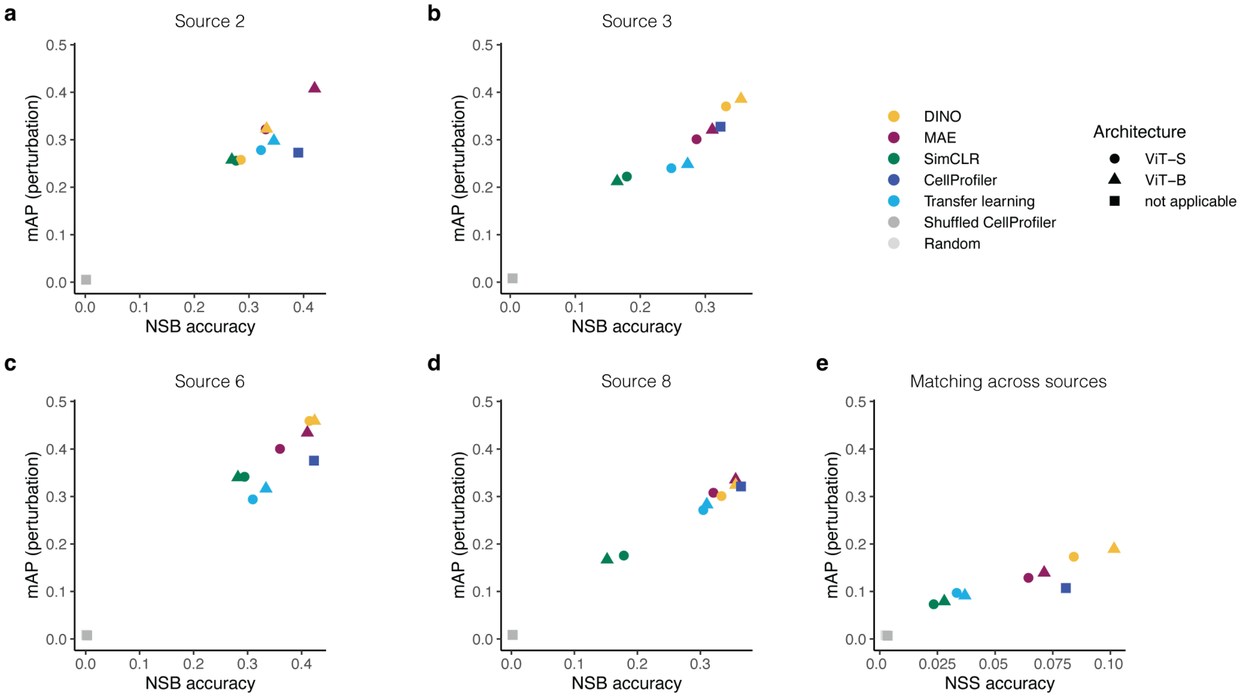


Extended Data Figure 3: F1-scores of SSL models on single-source and multisource evaluation sets. F1-score curves for matching perturbation (left) and target (right) labels for a range of nearest neighbors *k.* SSL models were trained on the multisource training set. Colors indicate different models and two randomized baselines: Shuffled CellProfiler (CellProfiler features with shuffled labels) and Random (random normally distributed features).

a) Performance of ViT-S architectures on the single-source evaluation set.

b) Performance of ViT-S architectures on the multisource evaluation set.

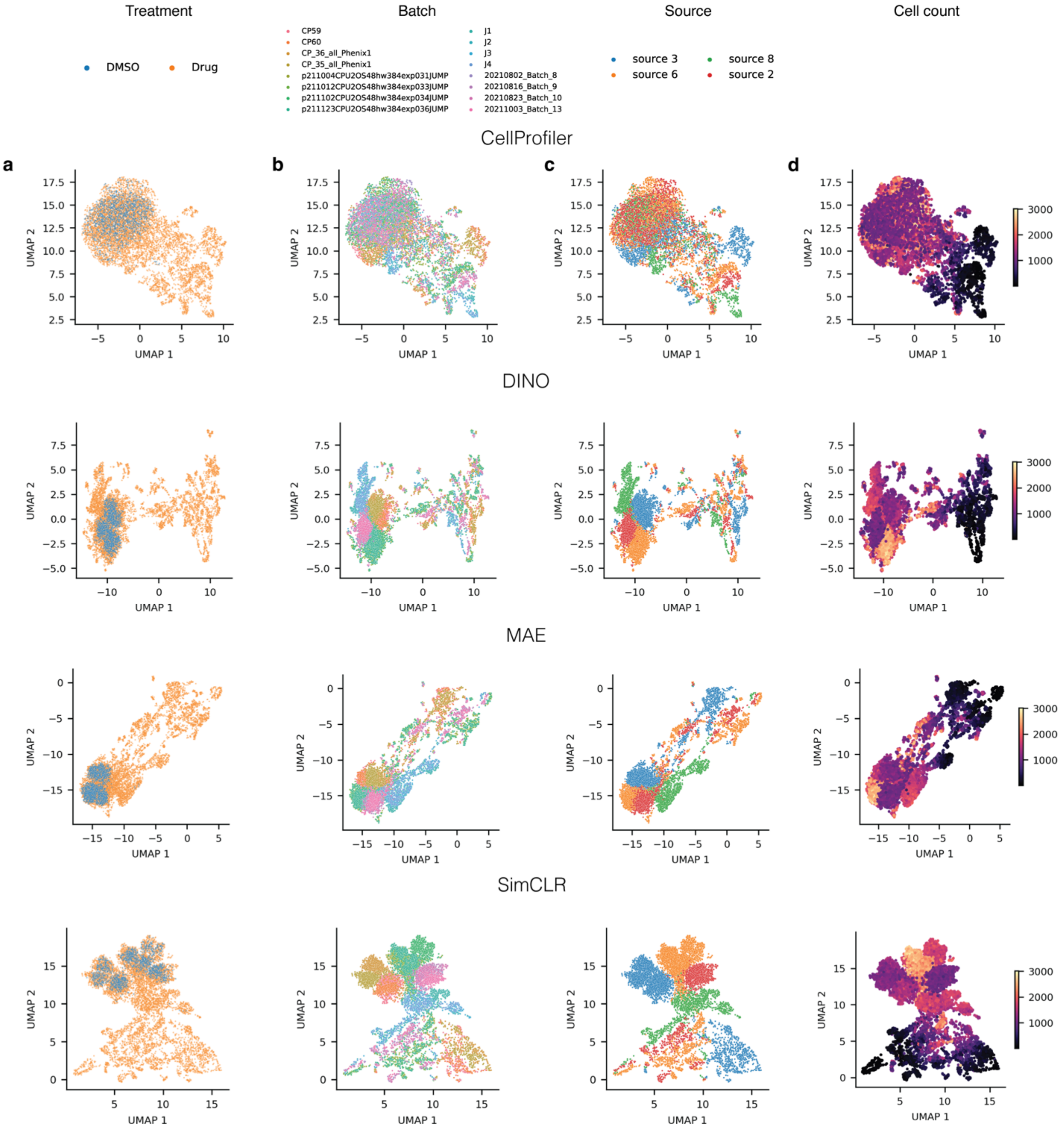
c) Performance of ViT-B architectures on the multisource evaluation set.



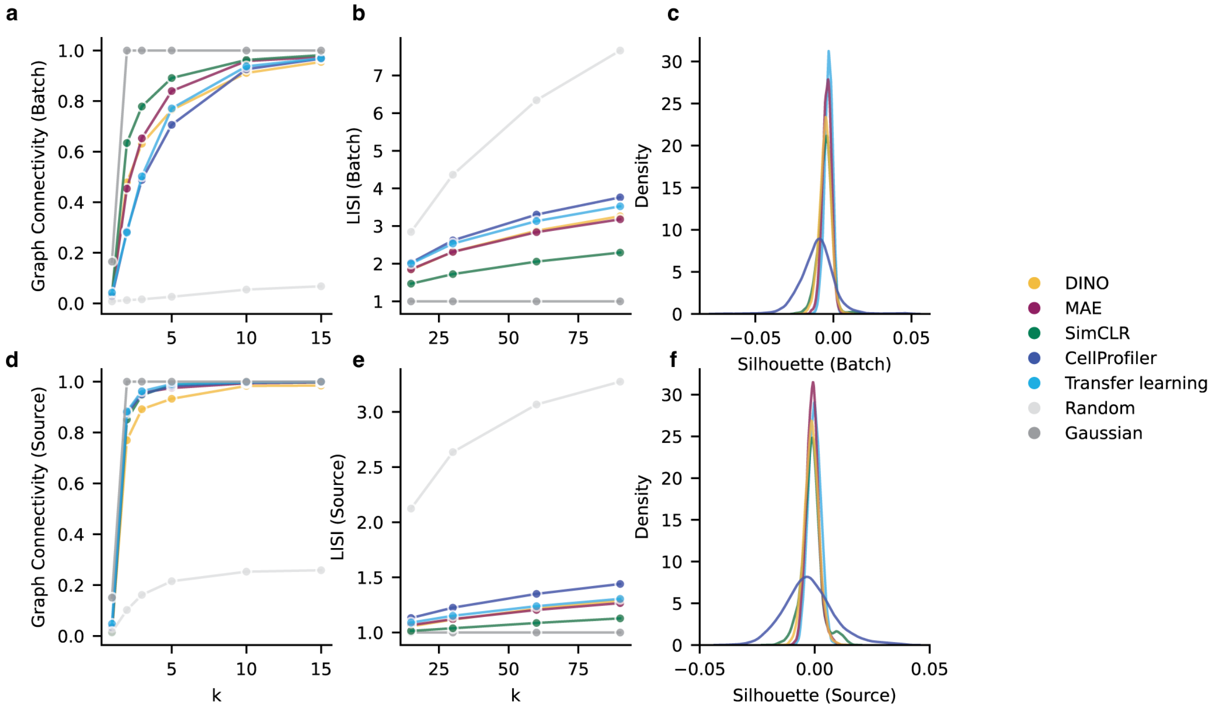
Extended Data Figure 4: Evaluation of SSL methods for individual JUMP-CP data sources. Perturbation reproducibility metrics computed on the multisource evaluation set for 4 JUMP-CP consortium data sources. SSL models were trained on the multisource training set. Shapes indicate the ViT architecture and colors indicate the model. Two randomized baselines were included: Shuffled CellProfiler (CellProfiler features with shuffled labels) and Random (random normally distributed features).

a)-d) Matching of perturbation labels for each individual JUMP-CP source.

e) Matching of perturbation labels across all 4 sources. *Mean average precision (mAP)* was computed on source-aggregated profiles. *Not-same-source (NSS) accuracy* quantifies the accuracy of matching well profiles across different data sources using a nearest neighbor classifier.

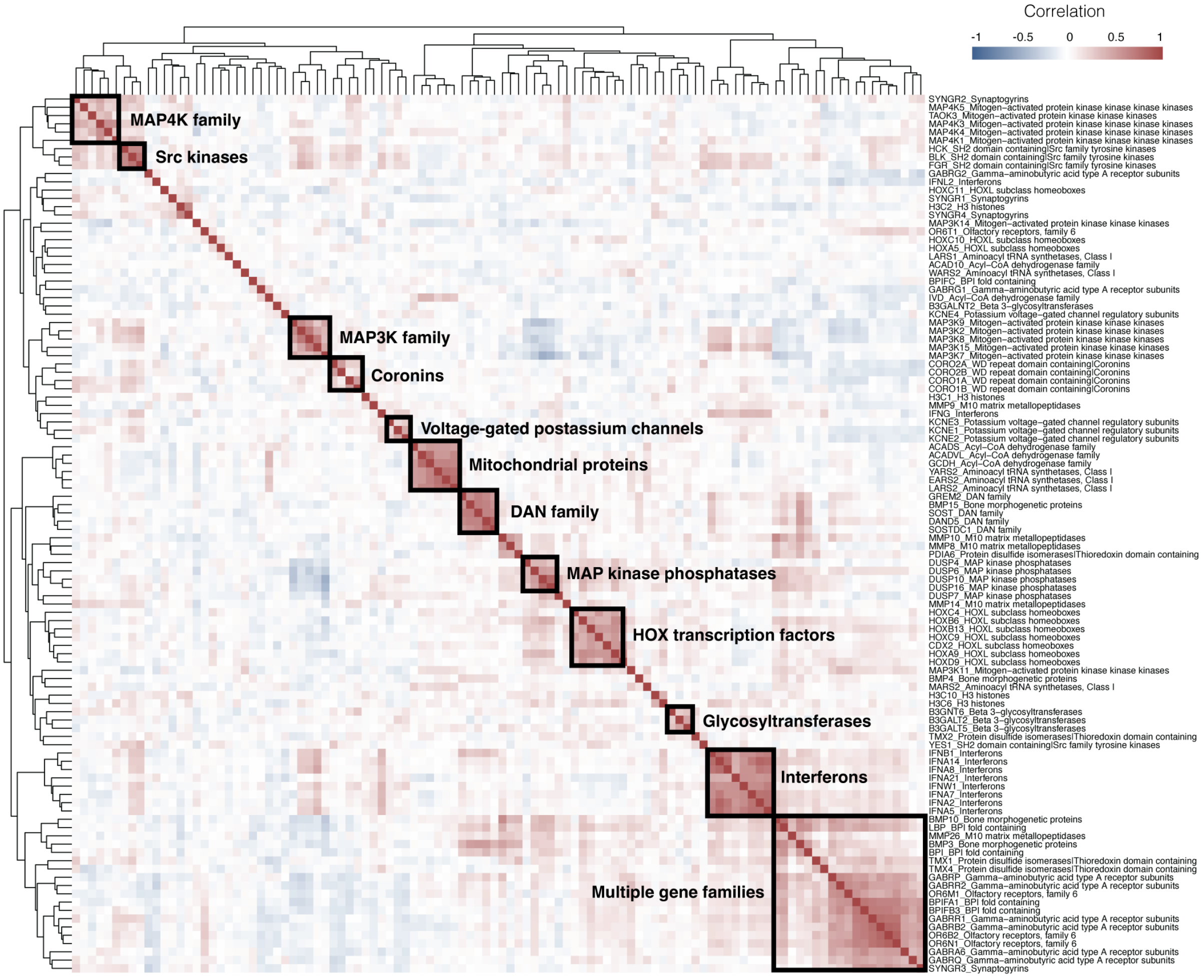


Extended Data Figure 5: UMAP projections of the multisource evaluation set reveal batch and source effects. UMAP projections of well representations for SSL models and the CellProfiler baseline. All SSL models were trained on the multisource training set with the ViT-S architecture. The points, corresponding to wells, are colored by a) treatment (DMSO negative control vs drug), b) batch, c) source, and d) cell count.

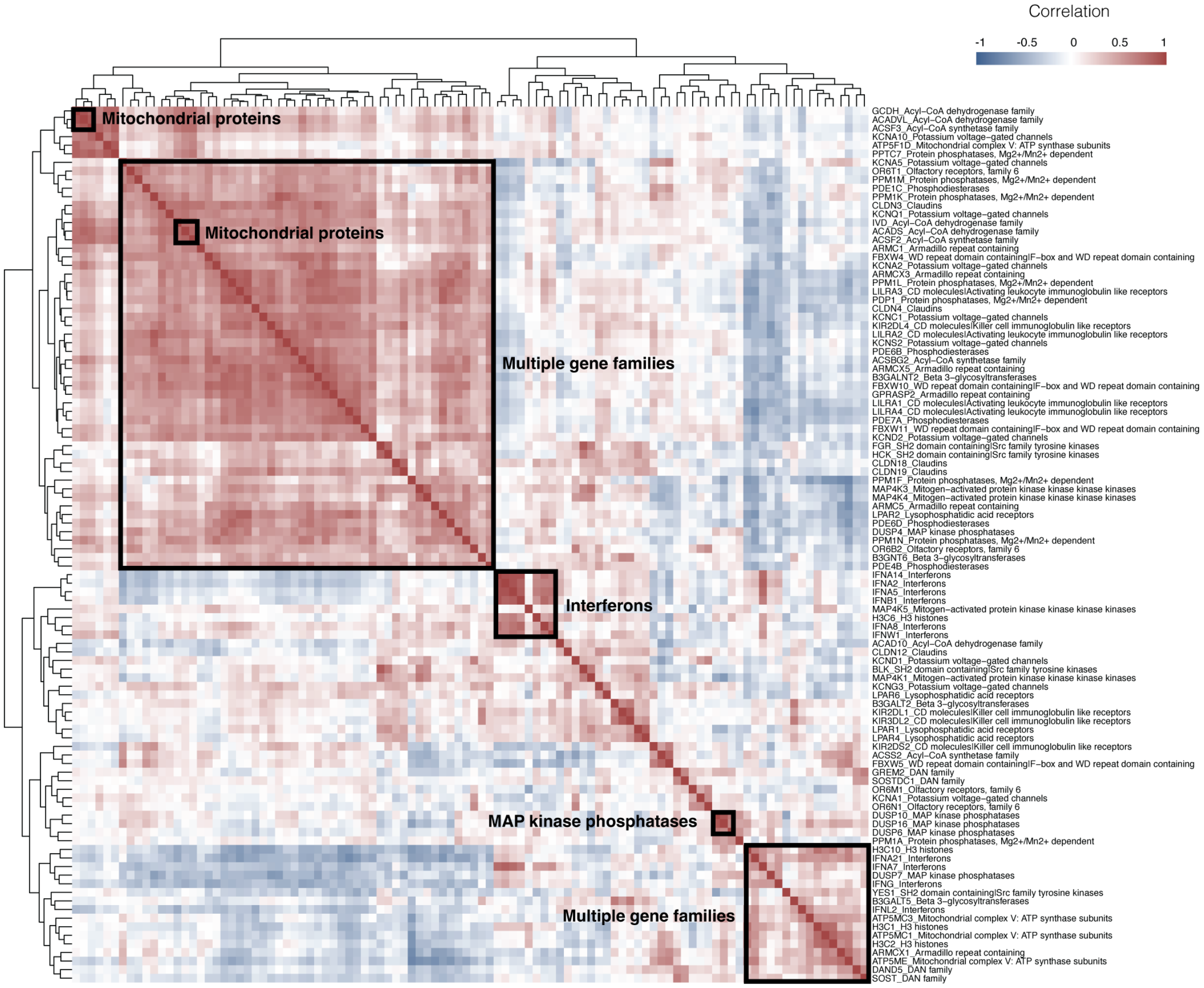


Extended Data Figure 6: Quantitative assessment of batch and laboratory effects for SSL and baseline methods.

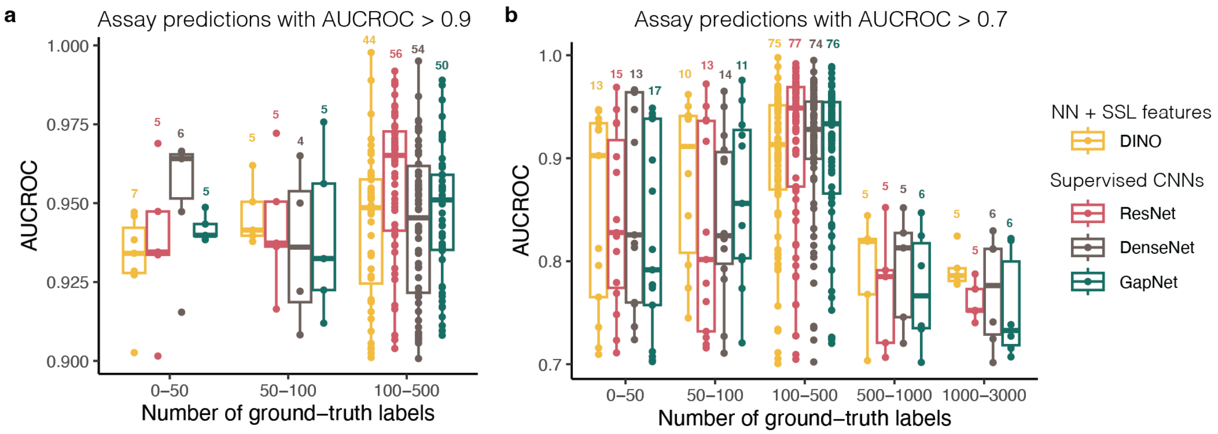
Impact of technical variation on SSL and baseline representations assessed on the multisource evaluation set. SSL models were trained on the multisource training set with the ViT-S architecture. Batch and laboratory (source) effects were assessed using graph connectivity, Local Inverse Simpson's Index (LISI), and silhouette scores. For reference, two synthetic representations were included: ‘Gaussian’ with non-overlapping Gaussian clusters and ‘Random’ with shuffled Gaussian cluster labels. **a)-c)** Metrics for batch effects. **d)-f)** Metrics for source effects.



Extended Data Figure 7: Hierarchical clustering of DINO representations for selected genetic perturbations. Hierarchical clustering of the pairwise correlation matrix of DINO representations for the subset of 20 gene families with the highest within-group correlations in the DINO feature space. Rows are labeled with perturbation and gene family names separated by an underscore (e.g. "HOXA9_HOXL subclass homeoboxes"). Clusters recapitulating gene groups, such as “HOX transcription factors” and “mitochondrial proteins”, are highlighted. DINO was trained on the multisource data with the ViT-S architecture.

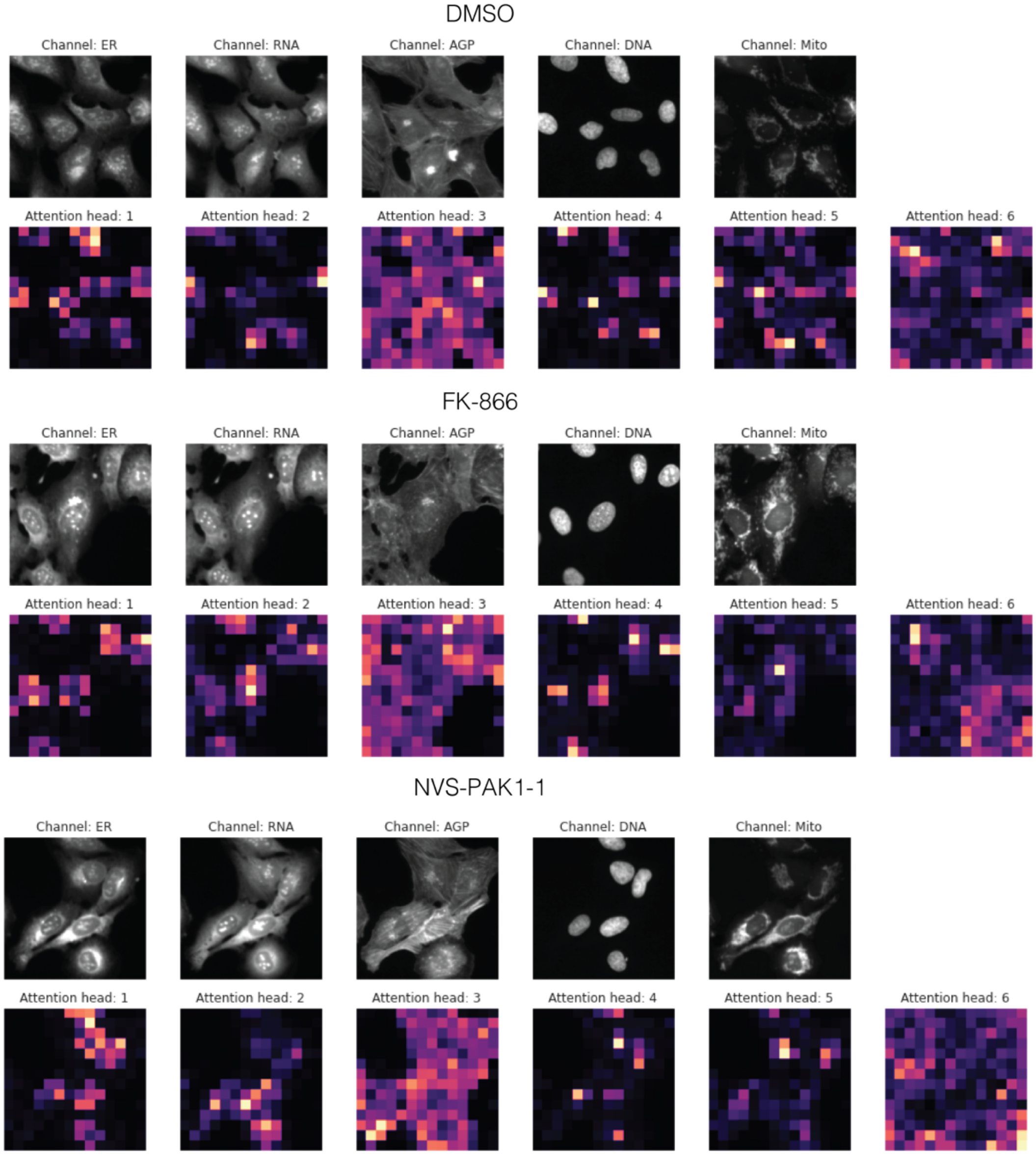


Extended Data Figure 8: Hierarchical clustering of CellProfiler representations for selected genetic perturbations.
Hierarchical clustering of the pairwise correlation matrix of CellProfiler representations for the subset of 20 gene families with the highest within-group correlations in the CellProfiler feature space. Rows are labeled with perturbation and gene family names separated by an underscore (e.g. “IFNA2_Interferons”). Clusters recapitulating gene groups, such as “mitochondrial proteins” and “MAP kinase phosphatases”, are highlighted.



Extended Data Figure 9: Performance of bioactivity prediction models stratified by assay data abundance. Compound activity prediction performance was compared between a neural network (NN) trained on DINO features and the top 3 convolutional neural networks (CNNs) from Hofmarcher et al.[2] across 209 ChEMBL assays grouped by the number of available activity labels (x-axis). Models were evaluated by the area under the receiver operating characteristic curve (AUCROC, y-axis). Assay AUCROC values are plotted for each model, with median and upper/lower quartile values summarized in boxplots. Only assays with **a)** AUCROC > 0.9 and **b)** AUCROC > 0.7 were considered. The numbers above boxplots indicate the assay counts per group.

[2] Hofmarcher, M., Rumetshofer, E., Clevert, D.-A., Hochreiter, S. & Klambauer, G. Accurate Prediction of Biological Assays with High-Throughput Microscopy Images and Convolutional Networks. *J Chem Inf Model* **59**, 1163–1171 (2019).



Extended Data Figure 10: DINO self-attention maps.
Cell Painting image crops and self-attention maps of the DINO attention heads in the last layer. Example images for DMSO, FK-866 and NVS-PAK1-1. The color scale in the self-attention maps represents the level of attention from the DINO `[CLS]` token, with lighter areas indicating higher attention. DINO was trained on the multisource data with the ViT-S architecture.

Extended Data Tables

Method	Input	Arch.	AUCROC	F1-score	AUC > 0.9	AUC > 0.8	AUC > 0.7
ResNet	Images	CNN	0.731 ± 0.19	0.508 ± 0.30	68	94	119
DenseNet	Images	CNN	0.730 ± 0.19	0.530 ± 0.30	61	98	121
GapNet	Images	CNN	0.725 ± 0.19	0.510 ± 0.29	63	94	117
DINO	**SSL features**	NN	**0.723 ± 0.18**	**0.507 ± 0.31**	**56**	**84**	**108**
MIL-Net	Images	CNN	0.711 ± 0.18	0.445 ± 0.32	61	81	105
M-CNN	Images	CNN	0.705 ± 0.19	0.482 ± 0.31	57	78	105
SC-CNN†	Images	CNN	0.705 ± 0.20	0.362 ± 0.29	61	83	109
CellProfiler†	Handcrafted features	NN	0.675 ± 0.20	0.361 ± 0.31	55	71	90

Extended Data Table 1: Performance comparison of bioactivity prediction models using Cell Painting. Eight deep learning methods were used to predict bioactivity labels for 209 ChEMBL assays using Cell Painting (see Methods). The methods included a neural network (NN) trained on DINO features, 6 convolutional neural networks (CNNs) trained directly on Cell Painting images, and an NN trained on CellProfiler features. Performance was evaluated on held-out test data, reporting means and standard deviations of AUCROC and F1-score values across all assays. Additionally, the number of assays with AUCROC above 0.9, 0.8, and 0.7 is reported for each method. The DINO results were obtained by applying a DINO model pretrained on the JUMP-CP data to the same dataset used in Hofmarcher et al.[3] The remaining results are from Table 1 of Hofmarcher et al.[3] Methods marked with † require cell-level segmentation, while the others use whole images as input. AUCROC: area under the receiver operating characteristic curve.

Method	Average time per plate	Processing time for 28 plates	AWS EC2 instance type	Cloud cost per plate
DINO	1.3 min.	36.6 min.	12 x g4dn.xlarge	$0.17
CellProfiler	66.7 min.	1867.2 min.	8 x r5.24xlarge + 4x r5.8xlarge	$10

Extended Data Table 2: Inference speed and cloud cost comparison for DINO and CellProfiler. Comparison of processing speed and cloud costs between DINO and CellProfiler pipelines for the analysis of 28 plates in the AWS cloud. The cloud costs include only EC2 instance charges.

[3] Hofmarcher, M., Rumetshofer, E., Clevert, D.-A., Hochreiter, S. & Klambauer, G. Accurate Prediction of Biological Assays with High-Throughput Microscopy Images and Convolutional Networks. *J Chem Inf Model* **59**, 1163–1171 (2019).