Spherical Phenotype Clustering

Luke Nightingale[1] Joseph Tuersley[1] Andrea Cairoli[1] Jacob Howes[1] Cameron Shand[1]
Andrew Powell[2] Darren Green[2] Amy Strange[1] Scott Warchal[1] Michael Howell[1]

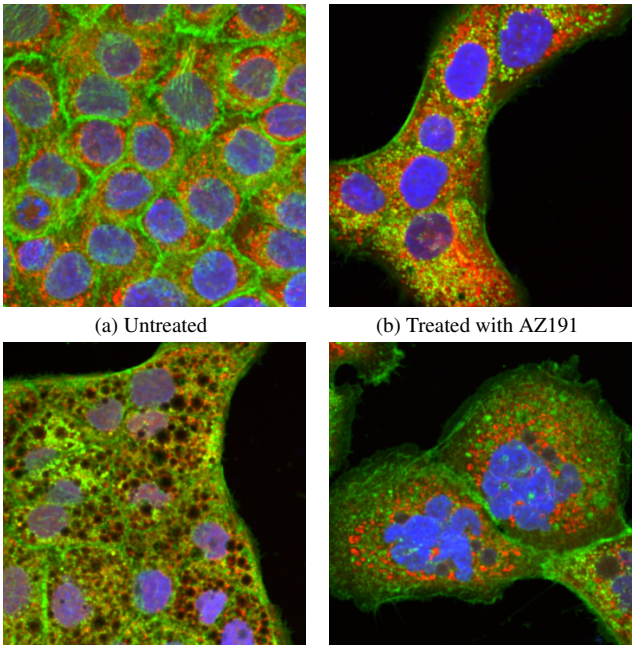
[1]The Francis Crick Institute, United Kingdom [2]GlaxoSmithKline, United Kingdom

Abstract

Phenotypic screening experiments comprise many images of the same cells perturbed in different ways, with biologically significant variation often subtle or difficult to see by eye. The specialized nature of the morphological changes and the fact that large quantities of data can be produced quickly makes training new machine learning models attractive. A byproduct of the experimental setup is knowledge of which well an image originated from and the treatment applied. We propose a geometrically inspired variant of contrastive learning incorporating this metadata. The method is tested on the BBBC021 benchmark dataset and in datasets of HaCaT cells perturbed chemically or genetically. On BBBC021 we attain higher NSC and NSCB scores than existing unsupervised (or weakly supervised) methods. In the HaCaT cells we find consistent quantitative gains and qualitative clustering reflecting underlying biology, including alignment of chemical and genetic perturbations.

1. Introduction

High-content screening (HCS) is a high throughput approach to drug discovery [2, 3]. To facilitate testing many biological samples in parallel, experiments are performed with test tubes called wells arrayed in a grid-like container known as a plate. The cells forming a sample are given perturbants and then subsequently imaged with a microscope. In this study a perturbation means either a chemical treatment consisting of a small molecule at a specified concentration, or a CRISPR gene knockout. Cells can be fluorescently tagged for proteins or structures such as nucleus and mitochondria, and these form a sequence of image channels. The data output of an experiment is a large number of multi-channel images accompanied by metadata indicating the perturbation, plate and well of origin. Some molecules also have a known mechanism-of-action (MoA). MoA describes the pharmacological interaction which is believed to underlie the effect of a compound. MoA induced phenotypes are often subtle and require an algorithm to distinguish, but in some cases are visible by eye (Figure 1). Alongside perturbed samples, each plate includes controls expected to elicit no response, typically DMSO in compound screens and non-targeting CRISPR reagents for genetic screens.



(a) Untreated (b) Treated with AZ191

(c) Treated with APY0201 (d) Treated with AMG900

Figure 1. HaCaT cells can exhibit visible cellular changes on treatment. Green channel: plasma membrane & F-actin; red channel: mitochondria; blue channel: DNA.

A common way to quantitatively analyse cellular images is with the software tool CellProfiler [6]. CellProfiler segments cells and extracts a large number of morphological measurements, for example cell area, nuclear eccentricity, mean mitochondrial intensity. This provides a feature vector for each cell, which can then be averaged over image or well. These are postprocessed and passed through a dimensionality reduction tool such as t-SNE [39]. The low dimensional plot is used to distinguish treatments from controls, discover outliers and infer unknown MoA. Handcrafted CellProfiler

features are one representation of cells that is suitable for downstream analysis, but with the advent of highly accurate deep learning algorithms, a promising application is the substitution of CellProfiler with deep learning based extraction of feature (embedding) vectors. Most closely analogous to CellProfiler is unsupervised learning, meaning MoA is not used to fit the algorithm and resulting image similarity can be thought of as intrinsic to the image content. Various approaches have been tried including transfer learning [1], contrastive learning [34], autoencoders [23] and weakly supervised learning [4].

Identically treated cells exhibit heterogeneity across a population, leading to substantial variation in shape and texture even within a well (Figure 1). Experimental metadata provides an opportunity to instruct an algorithm to learn features more representative of inter-well and inter-treatment variation. Weakly supervised learning (WSL) exploits this metadata by using it to train a classifier, typically on treatment condition. It continues to be a popular choice [21, 30].

Our goal is to make inferences about molecules based on reference perturbations, so that the desired algorithm discovers phenotypic clusters. Since the objective of a classifier is to distinguish classes, it is not clear that it provides an ideal objective for aligning similar perturbations. This study explores WSL in a contrastive setting. In conventional contrastive learning images are compared with their augmentations under the cosine similarity metric. We adapt several popular contrastive loss functions [9, 18] to compare images with experimental metadata and propose an update mechanism imposing less a-priori separation of treatments. We also suggest a new approach to batch correction.

2. Spherical Phenotype Clustering

The conventional way to incorporate metadata during training is with supervised learning using metadata labels as targets. For example a well or treatment classifier. We summarize the main components as follows:

- A dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ where each $\boldsymbol{x}_i$ is an image and N is the total number of images in the dataset. For each image there is an associated label y_i, an integer value between 1 and the total number of classes C.
- A parameterized encoder function f (the model) that takes an image and outputs a D dimensional embedding vector, $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$. In this work we use a ResNet18 [15] with an MLP projection head as the encoder function.
- A final linear layer that maps embedding vectors to class predictions comprising a weight matrix $W \in \mathbb{R}^{D\times C}$ and bias term $\boldsymbol{b} \in \mathbb{R}^C$, so that the predicted class is given by the maximum element of the vector $W^T\boldsymbol{z}_i + \boldsymbol{b}$.

From now on, we let dataset indexed quantities such as $\boldsymbol{x}_i$ instead be indexed by minibatch. During training the function f is optimized by repeatedly sampling a minibatch of M images from the dataset and updating parameters based on the gradient of the cross entropy loss function

$$l_i = -\log \frac{\exp(W_{y_i}^T \boldsymbol{z}_i + \boldsymbol{b}_{y_i})}{\sum_{j=1}^{C} \exp(W_j^T \boldsymbol{z}_i + \boldsymbol{b}_j)}, \tag{1}$$

where W_j indicates the jth column of W and can be viewed as a prototype embedding for class j [1]. In this and the following expressions we have omitted that the loss is averaged over batch samples, so that the actual loss is $L = (1/M)\sum_{i=1}^{M} l_i$.

Since treatments are always known when conducting a biological experiment, the usefulness of f in HCS is not in the direct ability to perform treatment classification, but rather being able to cluster similar treatments to deduce relationships between them. Following training, images are converted to vectors using f and then aggregated (by computing the mean) to create a well or treatment level profile, which is a vector of the same dimension as $\boldsymbol{z}_i$. The profiles are then compared with distance based metrics to assess how effectively they cluster according to a biologically meaningful label such as MoA. This is the sense in which the approach is described as weakly supervised - the metadata provides a weak subclass label which helps to learn the biological superclass a perturbation belongs to.

The first loss variant we consider is normalizing vectors to unit radius and including a temperature parameter. Defining the following similarity function

$$\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(\frac{1}{\tau}\frac{\boldsymbol{u}\cdot\boldsymbol{v}}{|\boldsymbol{u}||\boldsymbol{v}|}\right),$$

which is the exponential of the cosine similarity multiplied by the reciprocal of temperature τ, we have the L_2-constrained Softmax Loss [35, 41]

$$l_i = -\log \frac{\mathrm{sim}(\boldsymbol{z}_i, W_{y_i})}{\sum_{j=1}^{C} \mathrm{sim}(\boldsymbol{z}_i, W_j)}. \tag{2}$$

The normalized embeddings lie on a unit $(D-1)$-sphere. Optimizing over this space implicitly means learning angular relationships between embeddings. One argument given over the unnormalized softmax in Eq. (1) is that it is undesirable to use a radially sensitive optimization objective during training when downstream test metrics based on cosine distance ignore the radial component of an embedding [41]. This is the case in HCS, where cosine distance is used extensively in nearest-neighbour derived metrics and low dimensional plots generated with t-SNE or UMAP. In other work it is observed that when a class is tightly clustered on a hypersphere, the embeddings lie on a polar cap, which is linearly separable from the rest of the embeddings [43].

[1] The inner product grows as the angle between vectors decreases. Absence of radial information in the unit normalized cases that follow makes identification as class prototype embedding clearer.

The reciprocal of the temperature, $1/\tau$, has sometimes been viewed as a radial hyperparameter, controlling the amount of surface area available for clustering [35].

2.1. Contrastive options for incorporating experimental metadata

The contrastive loss in popular algorithm SimCLR is called NTXent [9]. Defining an augmented version of an image in the dataset as $\boldsymbol{x}_i^a$ and the corresponding embedding $\boldsymbol{z}_i^a = f(\boldsymbol{x}_i^a)$ we have the following expression for NTXent:

$$l_i = -\log \frac{\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_i^a)}{\sum\limits_{j=1}^{M} [\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j^a) + \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_{j\neq i})]},$$

where the notation $\boldsymbol{z}_{j\neq i}$ indicates the ith term is not included in the summation. This loss function encourages the embedding for an image to be closer to its augmentation than to embeddings for other images sampled from the dataset.

To introduce experimental metadata into NTXent we replace the augmented version of an image with the class vectors used in the loss functions above:

$$l_i = -\log \frac{\text{sim}(\boldsymbol{z}_i, W_{y_i})}{\sum\limits_{j=1}^{M} [\text{sim}(\boldsymbol{z}_i, W_{y_j}) + \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_{j\neq i})]}. \quad (3)$$

We see a close relationship with Eq. (2), where the summation now runs over the elements in a minibatch rather than over the classes, and there is also the inclusion of a term performing instance-instance comparisons.

A summation running over M rather than C elements is of interest in HCS where the number of classes in a dataset can be very large. For example, if they had decided to use weakly supervised learning with wells as targets, a recent work [21] would be training on the equivalent of $C = 93{\cdot}10^6$ classes. More generally the number of classes grows linearly with the size of the dataset. If we assume a (fixed) k images per well, the number of wells is N/k. Treatments similarly grow with the size of the dataset, but are usually performed with multiple replicates r, so that we would have N/kr treatment classes. Example values of k and r would be 160 and 12 for the HaCaT JUMP dataset described below.

The other contrastive loss function we consider is SupContrast [18]

$$l_i = \frac{-1}{\alpha} \log \sum_{p\in P(i)} \frac{\text{sim}(\boldsymbol{z}_i, W_{y_p}) + \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_{p\neq i})}{\sum\limits_{j=1}^{M} [\text{sim}(\boldsymbol{z}_i, W_{y_j}) + \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_{j\neq i})]},$$

where $P(i)$ is the set of batch indices labelled with class y_i, $\alpha = 2|P(i)| - 1$ and we have again adapted the original expression by inserting W.

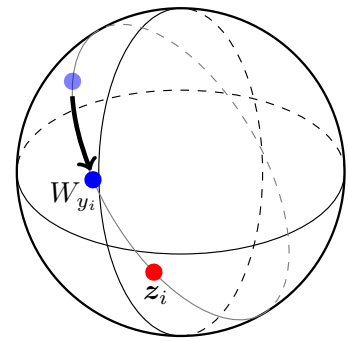


Figure 2. Geometric origin of class updates

Since they normalize vectors, the contrastive objectives also inherit the hyperspherical embedding distribution described in the previous section.

2.2. Spherical update paths

It has been observed in previous work [14, 41, 46] that loss functions on spheres of the type considered above can exhibit convergence issues related to the unnormalized embeddings failing to lie on the unit sphere. In [14] for example, it is shown that the gradient is orthogonal to a sphere, and then proven that the intuitive expectation finite step sizes along this gradient would lead to continually increasing embedding size holds and can slow convergence. It is also not clear that model parameters, which in the case of a ResNet are primarily filter kernel weights, and class vectors which play a role as embedding centroids, should be optimized under the same mechanism.

Motivated by this we consider the effect of optimizing class vectors by moving them along shortest spherical paths (geodesics) of the hypersphere. In order to define this trajectory we need a point for the class vector W_{y_i} to move towards, and the obvious choice is $\boldsymbol{z}_i$ a model generated embedding for class y_i. Even with random initialization of W, optimization of the model quickly makes the $\{\boldsymbol{z}_i\}$ non-random. We now observe that a closer study of this mechanism yields the following result:

The tangent to the shortest spherical path between W_{y_i} and $\boldsymbol{z}_i$ is a negative multiple of the gradient of the cosine similarity loss function.

A derivation of this relationship is given in Appendix A. Expanding the logarithm in NTXent Eq. (3) we have

$$\begin{aligned} l_i &= -\log \frac{\text{sim}(\boldsymbol{z}_i, W_{y_i})}{\sum_{j=1}^{M} [\text{sim}(\boldsymbol{z}_i, W_{y_j}) + \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_{j\neq i})]} \\ &= -\frac{1}{\tau} \frac{W_{y_i}^T \boldsymbol{z}_i}{|W_{y_i}^T||\boldsymbol{z}_i|} + \log \sum_{j=1}^{M} [\text{sim}(\boldsymbol{z}_i, W_{y_j}) + \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_{j\neq i})] \end{aligned}$$

The first term is the cosine similarity loss multiplied by a constant factor, referred to as the attractive component of the loss because it decreases as W_{y_i} and $\boldsymbol{z}_i$ draw close. Similarly, the second term has a repulsive effect. By the repeat appearance of the cosine loss, it is seen that spherical trajectories are closely related to gradient based optimization. We summarize the differences:

1. The arc path is confined to a specified radius, whereas gradient based updates move off the sphere.
2. A separate learning rate and optimizer configuration for model parameters and class vectors. For example momentum may be desirable for model parameters but distort the spherical update path.
3. Repulsive terms are included in the model optimization objective, but not class vector updates.

Item 3 has a naive attraction in the clustering context. Many classes, whether well or treatment, would be expected to be very similar, and leaving out repulsive terms would allow them to pack together so that similar classes exhibit a greater degree of overlap. Conversely including the repulsive terms may lead to a model that separates treatments of the same MoA.

The correspondences listed above allow us to implement the proposed mechanism with straightforward modifications to a conventional setup, and isolate the effect of each item. The model is optimized with Adam [20], but for the classes we use SGD with no momentum. Class vectors are optimized with the cosine similarity objective, and normalized after each gradient update.

2.3. Batch effect correction

HCS datasets are subject to unwanted technical variation due to experimental setup. The issue is related to the wider machine learning challenge of data distribution shift, but is particularly acute in HCS where there are data subgroups (batch) corresponding to the date an experiment was performed, and the plate samples were prepared on. Previous work [1, 4, 16, 34] has focused on postprocessing for the removal of batch effects. However, this does not correct clear separation of similar or identical treatments by batch (see for example Supplementary Figure 2B in [1]).

A recent approach to this problem is found in [26]. Along the lines of previous computer vision work [24] the authors propose unfreezing the batch normalization layers of a ResNet during evaluation, and instead computing the batch normalization statistics on the experimental batch (i.e. plate or experiment). We have found success with the idea of adapting batch norm layers, but in order to avoid normalizing out the phenotype in cases where plate and treatment are correlated, we implement a slightly different strategy. After training, for each plate we compute the batch norm layer statistics on the controls of that plate and then evaluate on the rest of that plate's samples. This is straightforward to implement in PyTorch [33] by using a BatchNorm layer with momentum 1.0. At evaluation time we perform the forward pass on plate controls while in train mode before freezing the BatchNorm layer by entering evaluation mode and then completing a forward pass on the treated samples. The technique has the effect of normalizing the model's internal image representations based on control samples in a manner reminiscent of normalizing CellProfiler output with z-score statistics computed on controls.

3. Results

We report results on three datasets, the public BBBC021 dataset and two new datasets of HaCaT cells. For each dataset we split the images in to training and evaluation sets, with splits described in the sections that follow. After training all embedding vectors for evaluation images originating from a given well are averaged to create a single well level embedding profile. Similarly, embeddings for a treatment are averaged to create a treatment level profile. All metrics are computed on evaluation split profiles. In each case we use the batch correction method described in Section 2.3.

The two metrics we work with are k-nearest neighbours with $k = 1$ (1-NN) and mean average precision (mAP). In both cases we use cosine distance as the underlying distance metric. In this context, the precision and recall values for mAP are calculated at different thresholds of distance from a sample. See [17] for a detailed discussion of mAP in HCS. Scoring based on mAP has recently grown in popularity [17, 21, 30]. The mAP metric has the distinct advantage over 1-NN of quantifying proximity between all members of a given class, not just first neighbours. For example situations may arise where two clusters exist for an MoA, where first nearest neighbour metrics give similar scores upon merging the clusters together but mAP gives substantially lower numbers. Unless otherwise stated, in the following sections both 1-NN and mAP use MoA as ground truth labels.

3.1. BBBC021

For comparison with other approaches on a publicly available dataset we use BBBC021 [5, 27], available online at https://bbbc.broadinstitute.org/BBBC021. BBBC021 is comprised of MCF-7 cells treated with 113 small molecules at eight different concentrations. Among the full dataset there is a curated subset of 103 treatments annotated with MoA based on visibility of phenotypes and prior literature. Most prior work has focused on this portion of the data, which we also use for our own benchmarking.

We follow [4, 30] in partitioning cells so that 60% form training data, and we use the other 40% for evaluation. The plate organisation in BBBC021 makes plate holdout difficult to compare with previous work, so we defer reporting using this method to the HaCaT datasets described in Section 3.2.

Table 1. BBBC021 ablation results

Model / Class Loss	mAP	mAP	NSC	NSCB
Profile	*well*	*treat*	*treat*	*treat*
Loss function variants				
Softmax (WSL)	0.790	0.900	96.7%	92.0%
L_2-Softmax	0.803	0.882	93.2%	92.3%
NTXent	**0.892**	0.931	**97.4%**	**95.6%**
SupContrast	0.882	**0.935**	**97.4%**	95.2%
Cosine	0.284	0.402	20.7%	18.8%
Spherical updates				
NTXent / NTXent	0.890	0.934	97.0%	94.9%
NTXent / Cosine	**0.924**	**0.959**	**98.0%**	**96.7%**
SupCon / SupCon	0.894	0.932	97.5%	96.1%
SupCon / Cosine	0.900	0.948	97.4%	95.6%

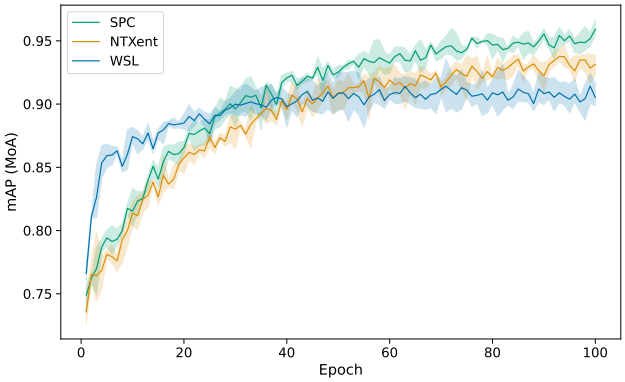


Figure 3. BBBC021 evaluation data, treatment profile MoA mAP during training with $\pm$ standard deviation across three runs.

The traditional way to report performance on BBBC021 has been under the NSC (Not-Same-Compound) and NSCB (Not-Same-Compound-or-Batch) scores on treatment level profiles [1, 4, 16, 28, 34, 37]. These are both 1-NN based metrics. In the NSC score, distances are only calculated between samples that have different compound labels, so that high scores require alignment between different compounds with the same MoA. NSCB additionally requires that neighbours come from a different batch.

3.1.1. Ablations

We begin by examining the effect of replacing the conventional WSL loss function with alternatives discussed in Section 2.1 - the two objectives with a full softmax, where the L_2 variant has normalized vectors and includes a temperature, and the two contrastive options. For the loss functions with a temperature hyperparameter we test three values, $\tau = 0.1, 0.25, 1$ and in all cases find $\tau = 0.25$ to give the best performance on MoA based metrics. We train a softmax based loss with and without the bias term but find limited difference, and report without the bias term in Table 1 for a consistent setup. Details of the other variants can be found in the supplementary material.

We find (Table 1) a substantial metric gain using either of the contrastive objectives. In particular the well aggregated mAP score is ≈ 0.1 higher than in the non-contrastive case, and the the treatment level mAP finds an increase > 0.03. Normalizing vectors and including the hyperparameter τ does not, by itself, result in improved metrics. However we note that metrics quantifying the ability to retrieve perturbation replicates can see an increase from the L_2 softmax variant. Indeed MoA metrics and replicate metrics do not always trend the same way. See the supplementary material for discussion of this phenomenon.

As described in Section 2.2, to test the effect of the spherical path modification to class vector optimization we normalize all class vectors after each optimization step and use two optimizers, Adam for the model parameters, and SGD with no momentum for the class vectors. All results in the second half of Table 1 are reported under this scheme. The cases where cosine similarity is used for class loss correspond to leaving out repulsive loss function terms.

Comparing the first and second parts of Tables 1, for both NTXent and SupContrast an increase can be seen using spherical paths, as long as we exclude repulsive terms. From here we continue with the NTXent / Cosine setup and refer to it as SPC. In Figure 3 we use intermediate model checkpoints to display progression of the mAP metric during training for SPC, WSL and (single objective) NTXent.

3.1.2. Improved benchmarks

Methods which use treatment metadata during training have described themselves as weakly supervised [4] or unsupervised [16]. In Table 2 we compare SPC to other results that did not use MoA during the fit. SPC gives improved NSC and NSCB scores comparative to extant methods. Comparing with the closest analogue, the treatment classifier [4], we find $+1\%$ NSC and $+10.2\%$ NSCB.

Some previous work on BBBC021 has focused on clustering the dataset to deduce similarity of treatments as opposed to developing a model that generalises to new data. To this end several methods use all (annotated) data during both training and evaluation [16, 34]. Fitting and evaluating SPC with all BBBC021 data results in an approximately $+1\%$ increase to NSC and NSCB over what we report in Table 2.

The MoA annotations for BBBC021 have been selected to give reliable ground truth. This supports benchmarking efforts, however HCS datasets often do not present classes that are as cleanly separable as BBBC021 or may not have semantic labels beyond treatment. In the next section we apply the method to data that has not been filtered other than to remove empty images, representing a more challenging, real-world scenario.

Table 2. BBBC021: comparison with previous work

Study	Type	NSC	NSCB
Ljosa *et al*. [28]	CellProfiler + Factor Analysis	94%	77%
Singh *et al*. [37]	CellProfiler	90%	85%
Lafarge *et al*. [23]	Autoencoder	92.9%	82.2%
Ando *et al*. [1]	Transfer Learning	96%	95%
Janssens *et al*. [16]	DeepCluster	97%	85%
Perakis *et al*. [34]	Contrastive Learning (SimCLR)	97%	94%
Caicedo *et al*. [4]	Treatment Classifier (DeepProfiler)	97%	86%
Ours	Contrastive Learning (SPC)	**98.0%**	**96.7%**

3.2. HaCaT datasets

To explore a diverse range of phenotypes generated in the cell painting assay, we adopt HaCaT cells, a spontaneously immortalised human keratinocyte cell line, and select two perturbant modalities: small molecules and genetic CRISPR knockouts. The JUMP-MoA small molecule library was initially developed by the Joint Undertaking for Morphological Profiling (JUMP) Cell Painting Consortium and aims at generating a large publicly available cell painting dataset [7]. This library comprises 90 compounds, 4 replicates per plate, covering 47 distinct MoA classes [11], and thus serves as a suitable choice to elicit a broad range of cellular phenotypes. For CRISPR perturbations we selected 207 genes anticipated to produce distinct phenotypes across the key cell painting channels, including those linked to mitochondrial function, Golgi apparatus, F-actin, and nuclear division/cell cycle. Chemical screening was performed at three compound concentrations (0.1, 1.0 and 10 uM) with three replicate plates per concentration, while the genetic screen was performed with 5 replicate plates. Images were acquired in 4 channels at 40X magnification, 10 fields of view per well.

Training and evaluation are carried out with three different approaches to test splits. In the first case, cells, we use a 60:40 image split similarly to the BBBC021 split. Then we hold back plates, one of each concentration for JUMP, and two of the five for CRISPR. Finally we hold back perturbations, for JUMP keeping back all 3 plates of 1uM treatments, and for CRISPR holding back a random 40% of KOs as well as 40% of control wells. In this way we verify the model can learn similarity between perturbations not seen during training.

We compare with CellProfiler and WSL. CellProfiler v4.2.5 is applied to segment cellular objects and measure approximately 1500 morphological features. This was followed by aggregating to a well median, feature selection, and robust z-scoring to the plate DMSO control median and median-absolute-deviation values. Except for the missing temperature, WSL is trained with the same hyperparameter configuration as SPC. On these datasets we find it advantageous to train SPC with $\tau = 0.1$ We report results in Table 3. Numbers for the JUMP dataset are a better estimate of model performance because we randomized well positions to account for the confounding effects of plate layout and ran multiple treatments for each MoA. For CRISPR we have only the KO label, so results reduce to perturbation metrics and furthermore we used a typical CRISPR screen layout with each perturbation represented by a single well per plate rather than four randomized positions. Nonetheless, in all cases SPC results in a clear improvement.

Table 3. HaCaT cells: 1-NN/mAP scores on well profiles

Method	Cells		Plates		Perturbations	
	1-NN	mAP	1-NN	mAP	1-NN	mAP
JUMP Data						
CellProfiler	0.327	0.115	0.271	0.113	0.330	0.196
WSL	0.482	0.164	0.398	0.152	0.461	0.240
SPC	**0.496**	**0.207**	**0.420**	**0.194**	**0.525**	**0.298**
CRISPR Data						
CellProfiler	0.550	0.197	0.522	0.246	0.755	0.243
WSL	0.626	0.326	0.545	0.318	0.772	0.370
SPC	**0.714**	**0.450**	**0.584**	**0.410**	**0.804**	**0.483**

3.2.1. Qualitative insights

We now train a model on both the JUMP and CRISPR datasets, using the cell train splits of each dataset, and investigate the evaluation profiles for relationships between perturbations. A t-SNE of well profiles (Figure 4) shows consistent clustering of multiple replicates for a given treatment, and also local structure such as concentration gradients and multiple compounds of the same MoA in adjacent mini-clusters. Some cluster arrangements violate MoA. In one we find three compounds marked with different MoA: PD-198306 (MAP kinase inhibitor), neratinib (EGFR inhibitor), selumetinib (MEK inhibitor). An explanation is offered by all three lying on the MAPK/ERK signalling pathway [32]. Inspection of the images gives marginal visual confirmation

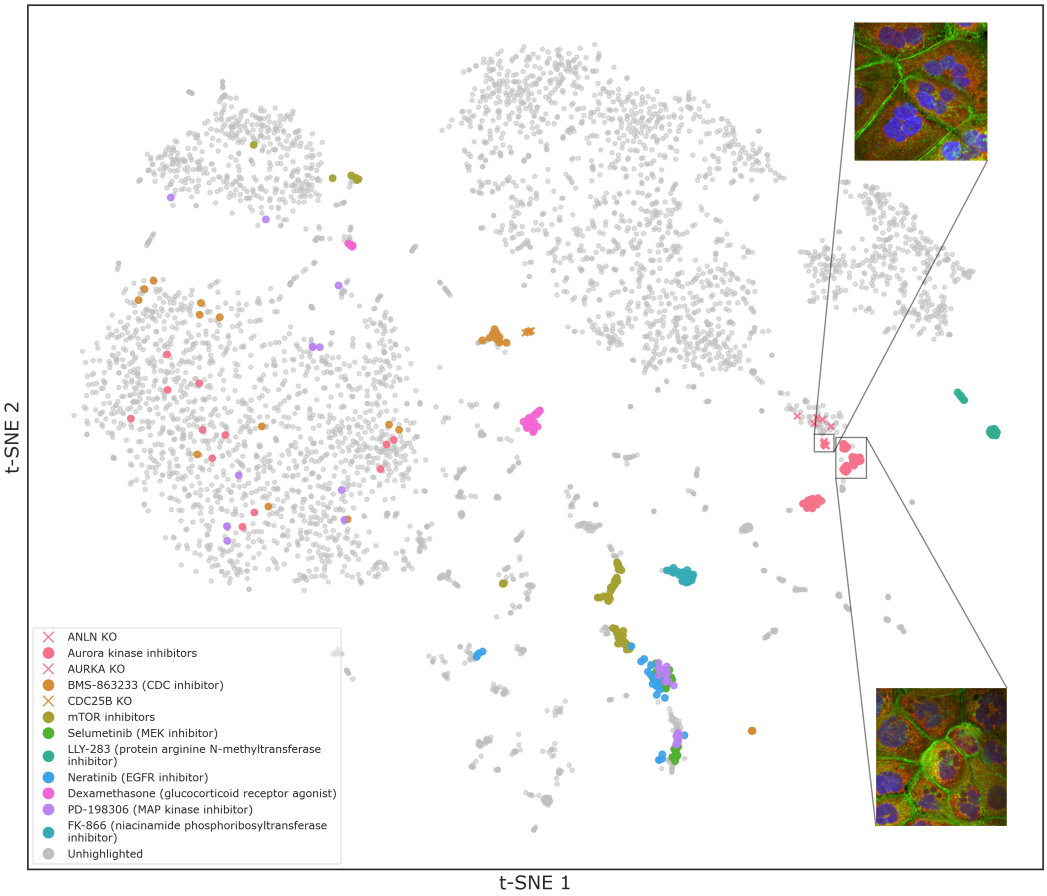


Figure 4. t-SNE of evaluation well profiles, with several arrangements highlighted. Highlighted unclustered wells are typically low concentration treatments. Insets: ANLN KO and AMG900.

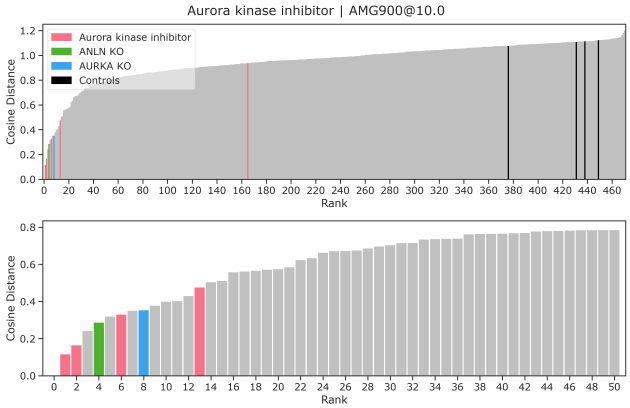


Figure 5. Ordered cosine distances between perturbation profiles and AMG900. First row: all profiles, second row: closest 50.

(Figure 6). The circumstance is clearer if we apply TGF-beta stimulation to the cells. Under these conditions the cells develop a characteristic staining pattern in the mitochondrial (red) channel. These observations suggest the algorithm is able to detect subtle cellular changes that visual inspection would agree with under more extreme conditions, and also illustrate the difficulty in effectively quantifying performance on datasets that are not curated in the manner of BBBC021.

Aurora kinase inhibitor compounds interfere with chromosome segregation during mitosis and have a very clear visual phenotype (Figure 1d). Examining the t-SNE we find clustering nearby CRISPR KOs of Aurora A kinase (AURKA) and ANLN. ANLN is also involved in mitosis, presenting a similar phenotype consisting of enlarged, multinucleated cells. To confirm the t-SNE is a faithful representation of model derived distances, we use perturbation profiles to compute the cosine distances between aurora kinase inibitor AMG900 and all perturbations in both datasets (Figure 5). The three unhighlighted perturbations closer than AURKA are CRISPR KOs of CCNA2, CDK1 and GOLGA2 which are closely related genes also involved in cell cycle and mitosis. For example GOLGA2 activates AURKA through its interaction with TPX2 [44]. It is seen that we

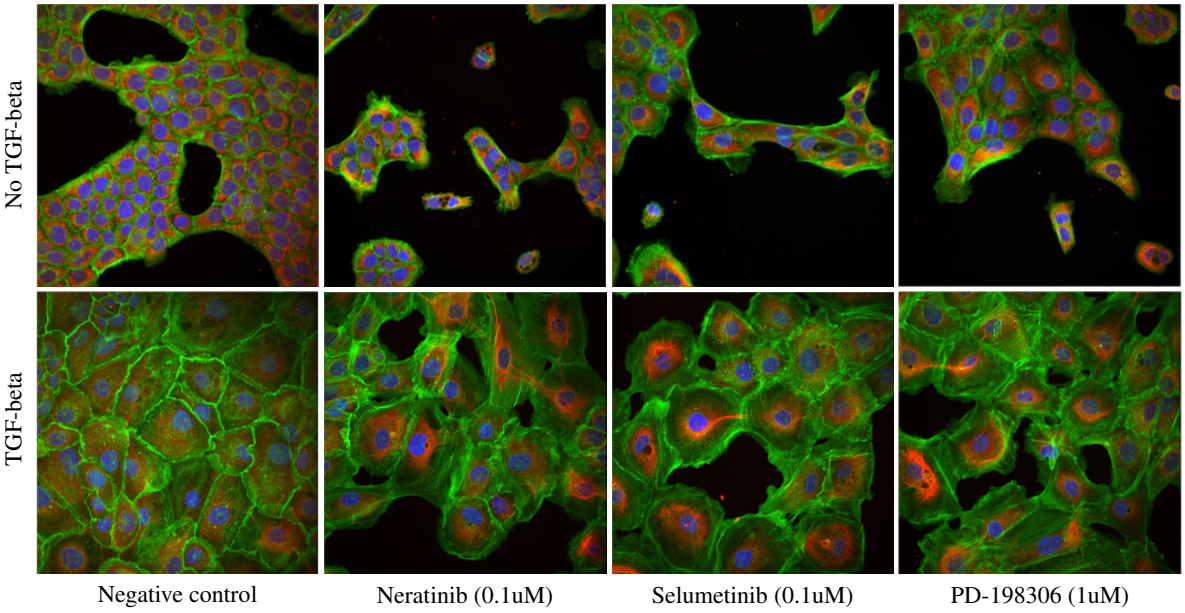


Figure 6. Visual similarity between three compounds with different MoA that SPC identified as similar, and negative controls for comparison. Cells are also shown with TGF-beta stimulation where the mitochondrial response becomes prominent.

are able to effectively deduce which gene KOs have similar effects to the AMG900 molecule. This type of tight clustering between chemical and genetic perturbations is an open challenge, and more generally even a small improvement in the ability to match chemicals to known phenotypic landmarks can greatly impact drug discovery by enabling the assignment of unknown chemical entities to a specific mechanism of action, target, or biological function [8].

4. Related Work

Contrastive learning is widely used in unsupervised representation learning. Popular realizations include SimCLR [9] and MoCo v2 [10]. Various methods implement image clustering in a contrastive setting. This can be explicit as with Contrastive Clustering [25] or implicit in the sense that the technique involves refining models that are trained with contrastive learning [31, 40]. SimCLR was previously adapted for phenotypic screening in [34] where it is applied to the BBBC021 dataset for state-of-the-art [22] results under the NSC metric. Applying contrastive learning in the labelled data setting is a motivating theme of SupContrast [18].

Spherical loss functions have been studied in facial recognition, from the L_2-softmax loss considered in this article [35, 41] to other adaptations involving margins and subclusters [12, 42]. In these works it is typically shown that the explicit spherical geometry outperforms a conventional softmax in image classification tasks. HCS and facial recognition share a link in both being supplied with large numbers of classes with limited representation. Approximating a softmax with a contrastive loss in the large class setting has been found useful or necessary in the past [29, 45]. The approach in [45], also a spherical loss, shows contrastive losses to be effective in the limit of each image having its own class vector representation, in this case the $> 10^6$ images of ImageNet.

Our approach is related to spherical interpolation which has a history in computer graphics [36]. NTXent, our loss function of choice, was named in [9] but appears in earlier work [38, 45] sometimes being referred to as NCELoss.

5. Conclusion

A notable feature distinguishing consumer images from those generated in HCS is some level of semantic labelling always being present in the form of experimental metadata. We have shown modifications to de-facto standard metadata based approach WSL which result in improved biological clustering. By avoiding training with MoA and injecting only the chemical or genetic content of a well, we inherit the desirable property of WSL that the method is a viable approach in any arrayed screen regardless of how well annotated the content. While this study has focused on moderate sized datasets, we expect future work to explore scaling and compare with large scale approaches using transformer backbones [13, 19, 21]. We also anticipate larger reference landscapes enabling the annotation of molecules of unknown mechanism with a variety of associations similar to those presented in this work.

References

[1] D Michael Ando, Cory Y McLean, and Marc Berndl. Improving phenotypic measurements in high-content imaging screens. *BioRxiv*, 2017. 2, 4, 5, 6, 1

[2] Marc Bickle. The beautiful cell: high-content screening in drug discovery. *Analytical and bioanalytical chemistry*, 398: 219–226, 2010. 1

[3] Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9):1757–1774, 2016. 1, 2

[4] Juan C Caicedo, Claire McQuin, Allen Goodman, Shantanu Singh, and Anne E Carpenter. Weakly supervised learning of single-cell feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9309–9318, 2018. 2, 4, 5, 6

[5] Peter D Caie, Rebecca E Walls, Alexandra Ingleston-Orme, Sandeep Daya, Tom Houslay, Rob Eagle, Mark E Roberts, and Neil O Carragher. High-content phenotypic profiling of drug response signatures across distinct cancer cells. *Molecular cancer therapeutics*, 9(6):1913–1926, 2010. 4

[6] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7:1–11, 2006. 1

[7] Srinivas Niranj Chandrasekaran, Jeanelle Ackerman, Eric Alix, D Michael Ando, John Arevalo, Melissa Bennion, Nicolas Boisseau, Adriana Borowa, Justin D Boyd, Laurent Brino, et al. Jump cell painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *bioRxiv*, pages 2023–03, 2023. 6

[8] Srinivas Niranj Chandrasekaran, Beth A Cimini, Amy Goodale, Lisa Miller, Maria Kost-Alimova, Nasim Jamali, John G Doench, Briana Fritchman, Adam Skepner, Michelle Melanson, et al. Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *Nature Methods*, pages 1–8, 2024. 8

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3, 8

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 8

[11] Beth A Cimini, Srinivas Niranj Chandrasekaran, Maria Kost-Alimova, Lisa Miller, Amy Goodale, Briana Fritchman, Patrick Byrne, Sakshi Garg, Nasim Jamali, David J Logan, et al. Optimizing the cell painting assay for image-based profiling. *Nature Protocols*, pages 1–44, 2023. 6, 2

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 8

[13] Michael Doron, Théo Moutakanni, Zitong S Chen, Nikita Moshkov, Mathilde Caron, Hugo Touvron, Piotr Bojanowski, Wolfgang M Pernice, and Juan C Caicedo. Unbiased single-cell morphology with self-supervised vision transformers. *bioRxiv*, pages 2023–06, 2023. 8

[14] Andrew Draganov, Sharvaree Vadgama, and Erik J Bekkers. The hidden pitfalls of the cosine similarity loss. *arXiv preprint arXiv:2406.16468*, 2024. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[16] Rens Janssens, Xian Zhang, Audrey Kauffmann, Antoine de Weck, and Eric Y Durand. Fully unsupervised deep mode of action learning for phenotyping high-content cellular images. *Bioinformatics*, 37(23):4548–4555, 2021. 4, 5, 6

[17] Alexandr A Kalinin, John Arevalo, Loan Vulliard, Erik Serrano, Hillary Tsang, Michael Bornholdt, Bartek Rajwa, Anne E Carpenter, Gregory P Way, and Shantanu Singh. A versatile information retrieval framework for evaluating profile strength and similarity. *bioRxiv*, 2024. 4

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 2, 3, 8

[19] Vladislav Kim, Nikolaos Adaloglou, Marc Osterland, Flavio M Morelli, Marah Halawa, Tim König, David Gnutt, and Paula A Marin Zapata. Self-supervision advances morphological profiling by unlocking powerful image representations. *bioRxiv*, pages 2023–04, 2023. 8, 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 2

[21] Oren Kraus, Kian Kenyon-Dean, Saber Saberian, Maryam Fallah, Peter McLean, Jess Leung, Vasudev Sharma, Ayla Khan, Jia Balakrishnan, Safiye Celik, et al. Masked autoencoders for microscopy are scalable learners of cellular biology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11757–11768, 2024. 2, 3, 4, 8

[22] Daniel Krentzel, Spencer L Shorte, and Christophe Zimmer. Deep learning in image-based phenotypic drug discovery. *Trends in Cell Biology*, 33(7):538–554, 2023. 8

[23] Maxime W Lafarge, Juan C Caicedo, Anne E Carpenter, Josien PW Pluim, Shantanu Singh, and Mitko Veta. Capturing single-cell phenotypic variation via unsupervised representation learning. In *International Conference on Medical Imaging with Deep Learning*, pages 315–325. PMLR, 2019. 2, 6

[24] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 4

[25] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8547–8555, 2021. 8

[26] Alexander Lin and Alex Lu. Incorporating knowledge of plates in batch normalization improves generalization of deep learning for microscopy images. In *Machine Learning in Computational Biology*, pages 74–93. PMLR, 2022. 4

[27] Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637–637, 2012. 4

[28] Vebjorn Ljosa, Peter D Caie, Rob Ter Horst, Katherine L Sokolnicki, Emma L Jenkins, Sandeep Daya, Mark E Roberts, Thouis R Jones, Shantanu Singh, Auguste Genovesio, et al. Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. *Journal of biomolecular screening*, 18(10):1321–1329, 2013. 5, 6, 1

[29] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in neural information processing systems*, 26, 2013. 8

[30] Nikita Moshkov, Michael Bornholdt, Santiago Benoit, Matthew Smith, Claire McQuin, Allen Goodman, Rebecca A Senft, Yu Han, Mehrtash Babadi, Peter Horvath, et al. Learning representations for image-based profiling of perturbations. *Nature communications*, 15(1):1594, 2024. 2, 4

[31] Chuang Niu, Hongming Shan, and Ge Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022. 8

[32] Richard J Orton, Oliver E Sturm, Vladislav Vyshemirsky, Muffy Calder, David R Gilbert, and Walter Kolch. Computational modelling of the receptor-tyrosine-kinase-activated mapk pathway. *Biochemical Journal*, 392(2):249–261, 2005. 6

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

[34] Alexis Perakis, Ali Gorji, Samriddhi Jain, Krishna Chaitanya, Simone Rizza, and Ender Konukoglu. Contrastive learning of single-cell phenotypic representations for treatment classification. In *Machine Learning in Medical Imaging: 12th International Workshop, MLMI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 12*, pages 565–575. Springer, 2021. 2, 4, 5, 6, 8

[35] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 2, 3, 8

[36] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985. 8, 1

[37] Shantanu Singh, M-A Bray, TR Jones, and AE Carpenter. Pipeline for illumination correction of images for high-throughput microscopy. *Journal of microscopy*, 256(3):231–236, 2014. 5, 6, 1

[38] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016. 8

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 1

[40] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020. 8

[41] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. 2, 3, 8

[42] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 8

[43] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020. 2

[44] Jen-Hsuan Wei, Zi Chao Zhang, R Max Wynn, and Joachim Seemann. Gm130 regulates golgi-derived spindle assembly by activating tpx2 and capturing microtubules. *Cell*, 162(2): 287–299, 2015. 7

[45] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 8

[46] Dingyi Zhang, Yingming Li, and Zhongfei Zhang. Deep metric learning with spherical embedding. *Advances in Neural Information Processing Systems*, 33:18772–18783, 2020. 3

Spherical Phenotype Clustering

Supplementary Material

A. Correspondence between cosine similarity loss and spherical arc

In the notation that follows $\hat{\boldsymbol{z}}$ denotes a unit vector in the direction of $\boldsymbol{z}$. A convenient expression for the path along a spherical arc is provided by graphics algorithm SLERP [36]. SLERP gives a spherically interpolated point between two vectors $\boldsymbol{w}$, $\boldsymbol{z}$ on a unit sphere, with $t \in [0, 1]$ indicating what proportion of distance along the arc the interpolated point should be

$$\boldsymbol{r}(t) = \frac{\sin[(1-t)\Omega]}{\sin\Omega}\hat{\boldsymbol{w}} + \frac{\sin[t\Omega]}{\sin\Omega}\hat{\boldsymbol{z}}, \qquad \text{(A.1)}$$

where Ω is the angle between $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{z}}$. It is a well known fact of high dimensional geometry that on an n-sphere, just as in the case of an ordinary sphere in three dimensional space, the shortest path between two points lies on a great circle. Great circles are defined by the intersection of the spherical surface with a 2-plane passing through the origin. On specifying a 2-plane using two points on the arc, the SLERP formula can be confirmed with straightforward geometry (Figure 7).

As described in the main article, the path we are interested in is the one starting at a class vector $\boldsymbol{w}$ and ending at a model embedding vector $\boldsymbol{z} = f(\boldsymbol{x})$ where image $\boldsymbol{x}$ has the same class as $\boldsymbol{w}$.

Proposition. *The tangent to the shortest spherical path between $\boldsymbol{w}$ and $\boldsymbol{z}$ is a negative multiple of the gradient of the cosine similarity loss function.*

The definition of the cosine similarity is $\cos\Omega = \boldsymbol{z}^T\boldsymbol{w}/|\boldsymbol{z}||\boldsymbol{w}|$ where Ω is the angle between $\boldsymbol{z}$ and $\boldsymbol{w}$. The cosine loss is the negative of this quantity. Computing the gradient of the cosine loss we get

$$\begin{aligned}\nabla_{\boldsymbol{w}}\left(-\frac{\boldsymbol{z}^T\boldsymbol{w}}{|\boldsymbol{z}||\boldsymbol{w}|}\right) &= -\frac{\boldsymbol{z}}{|\boldsymbol{z}||\boldsymbol{w}|} + \frac{\boldsymbol{z}^T\boldsymbol{w}}{|\boldsymbol{z}||\boldsymbol{w}|^3}\boldsymbol{w} \\ &= -\frac{\boldsymbol{z}}{|\boldsymbol{z}||\boldsymbol{w}|} + \frac{|\boldsymbol{z}||\boldsymbol{w}|\cos\Omega}{|\boldsymbol{z}||\boldsymbol{w}|^3}\boldsymbol{w} \\ &= -\frac{1}{|\boldsymbol{w}|}\left(\hat{\boldsymbol{z}} - \cos\Omega\,\hat{\boldsymbol{w}}\right). \qquad \text{(A.2)}\end{aligned}$$

Treating Eq. (A.1) as the path of a point over the sphere, we compute its tangent vector by taking the derivative with respect to t.

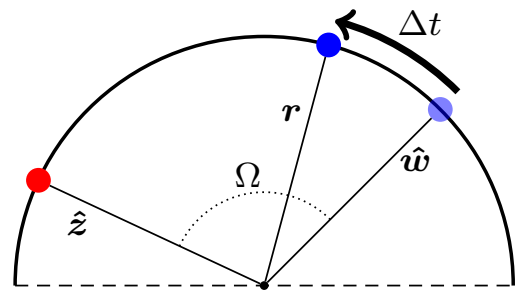


Figure 7. Verifying the SLERP formula

$$\begin{aligned}\left.\frac{d\boldsymbol{r}}{dt}\right|_{t=0} &= \frac{d}{dt}\left\{\frac{\sin[(1-t)\Omega]}{\sin\Omega}\hat{\boldsymbol{w}} + \frac{\sin[t\Omega]}{\sin\Omega}\hat{\boldsymbol{z}}\right\}\Bigg|_{t=0} \\ &= \frac{\Omega}{\sin\Omega}\Big(-\cos[(1-t)\Omega]\hat{\boldsymbol{w}} + \cos[t\Omega]\hat{\boldsymbol{z}}\Big)\Big|_{t=0} \\ &= \frac{\Omega}{\sin\Omega}\Big(-\cos\Omega\,\hat{\boldsymbol{w}} + \hat{\boldsymbol{z}}\Big),\end{aligned}$$

which is proportional to the negative of the gradient in Eq. (A.2) when $\Omega/\sin\Omega \geq 0$. This is the case since we use the smaller arc on a great circle giving $0 \leq \Omega \leq \pi$. □

B. Pre-processing and training details

We perform similar BBBC021 preprocessing to that in [1] including using the illumination correction functions [37], log transforming data and making use of the single cell crops defined in [28]. But we rescale channels independently and do not perform postprocessing, instead using the batch correction described in Section 2.3.

To preprocess the HaCaT datasets we resized each image to 512x512 pixels and generated 16 crops under a 4x4 tiling scheme (see Figure 8).[2] Since we have 10 fields per well the 4x4 tiling results in each well having 160 associated images of shape 128x128x4. We then further filter the dataset by removing any tiles with very limited cellular content. We do this by calculating an intensity threshold in the nuclear channel over the whole dataset (10% of the maximum minus minimum intensities, post outlier clipping). Tiles where less than 5% of the pixels pass the threshold value are removed from the dataset. We now rescale each channel independently to lie in a 0 to 1 range after first clipping outlier intensities on a per channel basis. The outlier

[2] In some experiments we find cells undergo significant size changes with treatment, making defining a consistent single cell crop resolution challenging. Tiling avoids this issue.

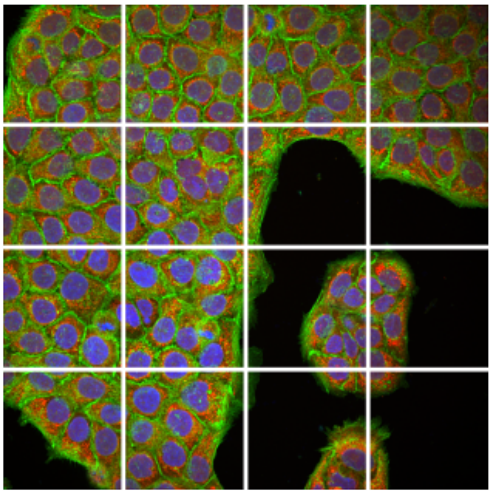


Figure 8. Raw 1080x1080 HaCaT images are resized to 512x512 and then tiled.

clipping thresholds we use are the 0.01 and 99.9 percentiles as in [19]. Images are converted to uint8 for storage.

All trained models use a ResNet18 architecture, with embedding dimension 384. Apart from batch correction comparisons in supplementary material, models used the customized batch norm layer described in Section 2.3. For BBBC021 we use the 3-channel augmentation scheme RandAugment with magnitude set to 12. For the 4-channel HaCaT cells we use three augmentations, RandomResizedCrop with scale (0.8, 1.0), RandomHorizontalFlip with probability 0.5, and GaussianBlur with kernel size 9, sigma 5, probability 0.5. We use the torchvision package to implement augmentations. Images in uint8 form are standardized by converting to float32, subtracting 127.5 from each pixel and then dividing by 255. Except for the combined JUMP+CRISPR model, training runs were terminated after 100 epochs without early stopping and used a batch size of 16. For the combined model we train 50 epochs and find improved integration with the larger batch size of 2048. Models were fit using the Adam optimizer [20] with an initial learning rate of 0.0003. SPC uses SGD with no momentum and learning rate 0.1 for the class vectors. All tabulated experiments were run three times, and the mean reported.

The majority of training and evaluation was performed on a 32GB memory NVIDIA V100 GPU with a 20-core Intel Xeon Gold 6148 CPU and 256GB RAM.

C. Further results

We collect in Table 4 extended results for BBBC021. Replicate (rep) based metrics quantify the extent to which the closest samples in a well profile's neighbourhood are other wells of the same treatment. An interesting phenomenon emerges of replicate metrics falling on increase of MoA scores. The qualitative explanation for what is taking place is that while high performing models for replicate retrieval pack multiple replicates together closely, those replicates are often no longer clustered with the rest of the MoA (Figure 9). Conversely, SPC and the other contrast objectives send replicates to a well defined MoA cluster, but within that cluster there is overlap with other treatments of the same MoA.

To some extent this behaviour can be calibrated with the temperature. Indeed, the highest scores on replicate metrics come from choosing a low temperature on the L_2-Softmax. Temperature shows itself to be closely related to achieving biologically meaningful clustering. We also compare WSL and SPC without performing any batch correction, and see the same performance improvement trends. Note that several MoAs appear in only a single batch and are hence not used in the NSCB calculation, which is one explanation for high uncorrected NSCB scores.

In Figure 4 we found that SPC gave convincing qualitative clustering of the glucocorticoid receptor agonist dexamethasone. Dexamethasone is an example of a compound for which it is challenging to see any visually discernible effect (Figure 10), and furthermore we have found limited success extracting it with CellProfiler. Quantitatively the mAP for this MoA is 0.14 using CellProfiler and 0.64 for the model of Figure 4 illustrating that our deep learning method can identify phenotypes that might otherwise be missed.

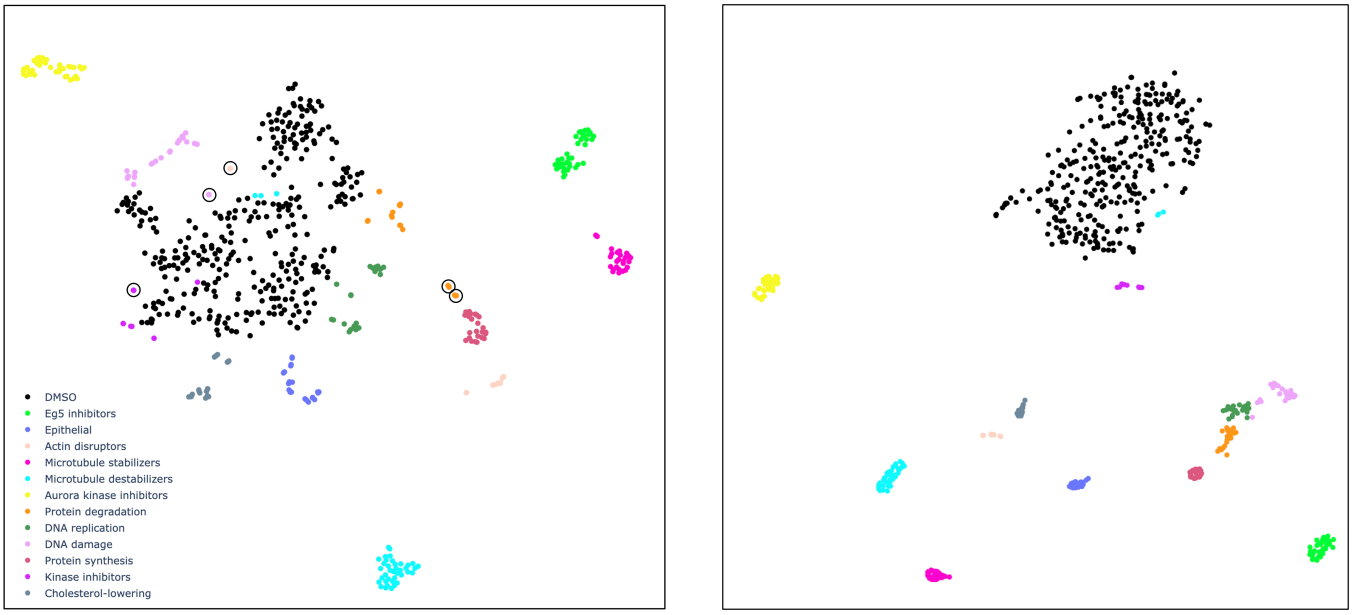
We give another presentation of the AMG900 results in Table 5. The JUMP and CRISPR experiments were performed on different plates weeks apart, so that we overcome clear experimental batches. Comparing the batch correction of Section 2.3 with commonly used postprocessing approach TVN reveals more convincing chemical-genetic clustering, absent a number of chemical MoAs interspersed. Performing no batch correction or using only the whitening component of TVN produces little to no chemical-genetic alignment.

D. HaCaT screening

We followed the standard cell painting assay protocol [3, 11], except for optimisations of dye concentrations to achieve optimal staining in HaCaT cells. For our staining procedures, MitoTracker was added live to cells, followed by cell fixation, permeabilisation and staining for the following subcellular compartments: nuclear DNA (Hoechst 33342), endoplasmic reticulum (concanavalin Alexa-488), RNA and nucleoli (syto-14), Golgi apparatus and plasma membrane (wheat germ agglutinin Alexa-555), and F-actin (phalloidin Alexa-568). Eight chemical substitutions to JUMP-MoA were made due to intellectual property constraints and two due to supply issues. We treated JUMP for 3 days. CRISPR screening was performed using 10 nM guide RNA, treating for 5 days to ensure maximum gene knockdown efficiency. Images were acquired using an Opera Phenix Plus high content microscope, and were max projected per channel.

Table 4. Extended BBBC021 results

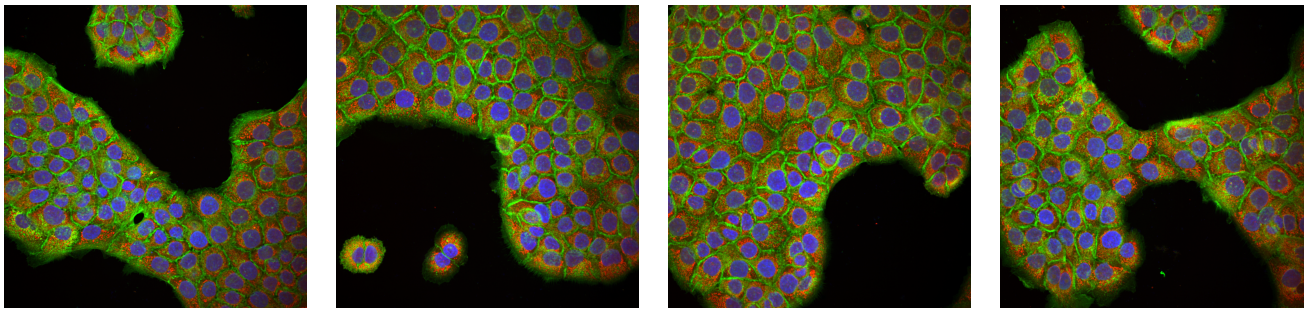
Variant	τ	MoA 1-NN	Rep 1-NN	Rep mAP	MoA mAP	MoA mAP	NSC	NSCB
Profile		*well*	*well*	*well*	*well*	*treatment*	*treatment*	*treatment*
Loss function variants								
(WSL) Softmax	-	0.995	0.818	0.669	0.790	0.900	96.7%	92.0%
Softmax w/ bias	-	0.994	0.812	0.657	0.792	0.911	94.4%	92.7%
L_2-Softmax	0.1	0.992	**0.931**	**0.847**	0.666	0.789	82.2%	78.6%
L_2-Softmax	0.25	0.988	0.701	0.496	0.803	0.882	93.2%	92.3%
L_2-Softmax	1.0	0.797	0.566	0.10	0.292	0.388	40.1%	30.7%
NTXent	0.1	0.989	0.709	0.495	0.824	0.882	91.2%	91.3%
NTXent	0.25	0.995	0.692	0.457	0.892	0.931	97.4%	95.6%
NTXent	1.0	0.981	0.646	0.373	0.841	0.858	91.2%	79.7%
SupContrast	0.1	0.992	0.697	0.462	0.826	0.894	93.8%	92.0%
SupContrast	0.25	0.993	0.686	0.453	0.882	0.935	97.4%	95.2%
SupContrast	1.0	0.971	0.633	0.345	0.785	0.822	91.2%	76.8%
Cosine	-	0.680	0.517	0.096	0.284	0.402	20.7%	18.8%
Spherical updates								
NTXent / NTXent	0.25	0.996	0.693	0.460	0.890	0.934	97.0%	94.9%
(SPC) NTXent / Cosine	0.25	0.996	0.669	0.425	**0.924**	**0.959**	**98.0%**	**96.7%**
SupCon / SupCon	0.25	0.993	0.700	0.476	0.894	0.932	97.5%	96.1%
SupCon / Cosine	0.25	0.992	0.662	0.409	0.900	0.948	97.4%	95.6%
No batch correction								
WSL	-	0.998	0.867	0.753	0.708	0.880	87.7%	92.0%
SPC	0.25	**1.000**	0.720	0.537	0.903	0.918	88.0%	96.0%



(a) L_2-Softmax ($\tau = 0.1$)

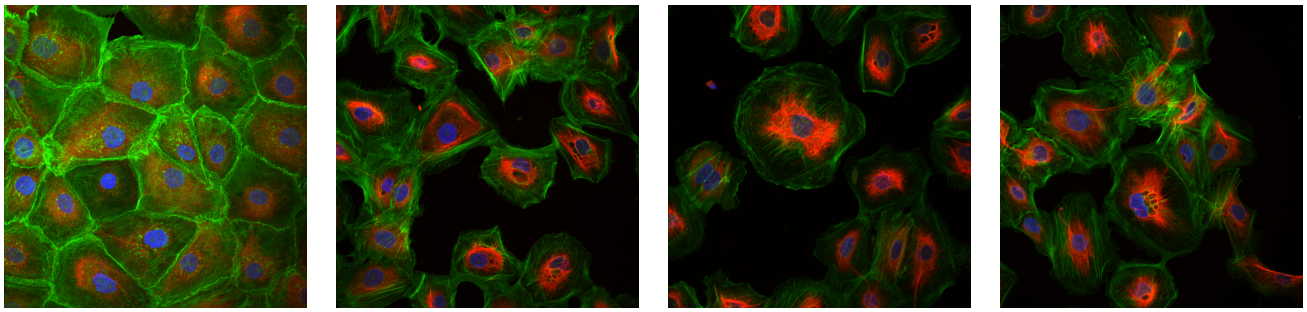
(b) SPC ($\tau = 0.25$)

Figure 9. High performance on replicate matching doesn't necessarily improve biologically meaningful clustering. Circled in the L_2-Softmax plot are instances where all treatment replicates are tightly packed, but detached from a main MoA cluster. By contrast SPC MoA clusters are well defined, even if there is internal overlap of treatments. Plots generated with TSNE on evaluation well profiles.



(a) Untreated (b) dexamethasone @ 0.1uM (c) dexamethasone @ 1uM (d) dexamethasone @ 10uM

Figure 10. Glucocorticoid receptor agonist dexamethasone is difficult to distinguish from control samples by inspecting images.



(a) Untreated (b) Neratinib @ 1uM (c) Selumetinib @ 1uM (d) PD-198306 @ 10uM

Figure 11. Increasing concentration in TGF-beta stimulated HaCaT cells causes the MAPK/ERK visual phenotype to become even more pronounced.

Table 5. AMG900@10uM top 25 closest perturbations: comparing TVN and proposed batch correction, KOs highlighted.

Rank	TVN			Proposed		
	Perturbation	MoA	Distance	Perturbation	MoA	Distance
0	AMG900@10.0	Aurora kinase inhibitor	0.00	AMG900@10.0	Aurora kinase inhibitor	0.00
1	AMG900@1.0	Aurora kinase inhibitor	0.26	AMG900@1.0	Aurora kinase inhibitor	0.11
2	MK-5108@10.0	Aurora kinase inhibitor	0.35	AMG900@0.1	Aurora kinase inhibitor	0.16
3	AMG900@0.1	Aurora kinase inhibitor	0.37	CCNA2 KO	CCNA2 KO	0.24
4	n/a@0.1	kinesin inhibitor	0.38	ANLN KO	ANLN KO	0.29
5	BX-912@10.0	pyruvate dehydro. kin. inhib.	0.38	CDK1 KO	CDK1 KO	0.32
6	n/a@1.0	kinesin inhibitor	0.43	MK-5108@10.0	Aurora kinase inhibitor	0.33
7	SCH-900776@1.0	CHK inhibitor	0.50	GOLGA2 KO	GOLGA2 KO	0.35
8	MK-5108@1.0	Aurora kinase inhibitor	0.52	AURKA KO	AURKA KO	0.35
9	n/a@10.0	kinesin inhibitor	0.61	BX-912@10.0	pyruvate dehydro. kin. inhib.	0.38
10	BLU9931@10.0	FGFR inhibitor	0.64	NDC80 KO	NDC80 KO	0.40
11	PF-477736@10.0	CHK inhibitor	0.66	WEE1 KO	WEE1 KO	0.40
12	SCH-900776@10.0	CHK inhibitor	0.68	PF-477736@10.0	CHK inhibitor	0.43
13	valrubicin@10.0	DNA inhibitor	0.52	MK-5108@1.0	Aurora kinase inhibitor	0.48
14	PD-198306@10.0	MAP kinase inhibitor	0.76	RPL12 KO	RPL12 KO	0.50
15	CDK1 KO	CDK1 KO	0.77	valrubicin@10.0	DNA inhibitor	0.51
16	ANLN KO	ANLN KO	0.78	CHEK1 KO	CHEK1 KO	0.56
17	olaparib@10.0	PARP inhibitor	0.80	KRT14 KO	KRT14 KO	0.56
18	NUP153 KO	NUP153 KO	0.80	CDC25B KO	CDC25B KO	0.56
19	CCNA2 KO	CCNA2 KO	0.81	n/a@10.0	kinesin inhibitor	0.57
20	neratinib@1.0	EGFR inhibitor	0.82	n/a@1.0	kinesin inhibitor	0.57
21	selumetinib@10.0	MEK inhibitor	0.82	olaparib@10.0	PARP inhibitor	0.58
22	CHIR-99021@10.0	glycogen synthase kinase inhibitor	0.84	BMS-863233@1.0	CDC inhibitor	0.62
23	acriflavine@0.1	hypoxia inducible factor inhibitor	0.84	neratinib@10.0	EGFR inhibitor	0.63
24	GOLGA2 KO	GOLGA2 KO	0.85	n/a@0.1	kinesin inhibitor	0.66
25	acriflavine@10.0	hypoxia inducible factor inhibitor	0.85	neratinib@1.0	EGFR inhibitor	0.67