Spherical Phenotype Clustering

Luke Nightingale[1], Joseph Tuersley[2], Andrea Cairoli[1], Jacob Howes[1], Andrew Powell[3], Darren Green[4], Amy Strange[1], Scott Warchal[2], and Michael Howell[2]

[1]Software Engineering and AI, The Francis Crick Institute
[2]High Throughput Screening, The Francis Crick Institute
[3]Crick-GSK Biomedical LinkLabs, GSK
[4]GSK Research & Development, GSK

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. This contrasts with consumer images which do not guarantee an associated categorisation. We propose a non-parametric variant of contrastive learning incorporating this metadata. The method is tested on the BBBC021 benchmark dataset and in HaCat cells treated with the JUMP reference compound set. On BBBC021 we attain higher NSC and NSCB scores than existing unsupervised (or weakly supervised) methods. In the HaCat cells we attain significantly better quantitative results (>10%) than CellProfiler or SimCLR and qualitative clustering reflecting underlying biology.

1 Introduction

High-content screening (HCS) is an approach to drug discovery where cells are treated with chemical compounds and subsequently imaged with a microscope. Cells can be stained for various proteins or structures such as nucleus and mitochondria and these form a sequence of image channels. Experiments are performed on plates, where each well is imaged multiple times in different spatial locations called fields. The data output of an experiment is a large number of multi-channel images accompanied by metadata indicating the treatment type, plate and well of origin as well as tags such as the mechanism-of-action (MoA). MoA provides the biochemical interaction which is believed to underlie the effect of a compound. It is expected that among the images certain morphological changes will be observable in the cells that are related to treatment and MoA (Figure 1). Along with the chemically treated samples, there is typically also included a number of negative controls.

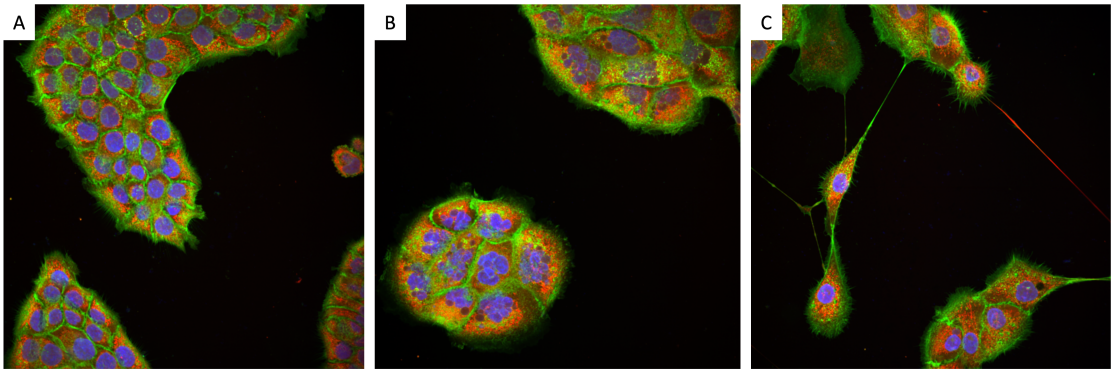

Figure 1: Chemical treatments elicit visible cellular phenotypes. A) untreated HaCat cells B) treated with AMG900 C) treated with Palbociclib.

A common way to quantitatively analyse images for the effect of compounds on cells is with the software tool CellProfiler [6]. CellProfiler segments cells and extracts a large number of morphological measurements. This provides a feature vector for each cell, which can then be averaged over image or well. These are postprocessed and then passed through a dimensionality reduction tool such as UMAP [26]. The UMAP plot is used to distinguish treatments from controls, discover outliers and infer unknown MoA.

CellProfiler's extraction of morphological measurements is one representation of cell images 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. A simple and widely used approach is transfer learning [1] where the penultimate (or other internal) layer from a model pretrained on consumer image datasets is used to generate embedding vectors. The vectors can then be compared and visualized to deduce treatment similarity. Other approaches based on autoencoders offer embedding vectors in the form of a latent space [20]. A method which offers high benchmark performance is DeepCluster [14]. Originally developed by Facebook [5], the technique involves iteratively clustering a dataset with k-means and using the k-means assignments as pseudolabels. Other deep learning based clustering techniques are often formulated within the broader unsupervised learning framework of contrastive learning. This can be explicit as with contrastive

clustering [22] or implicit in the sense that the technique involves refining models that are trained with contrastive learning [34, 28].

Contrastive learning is an approach to unsupervised learning where a machine learning model is required to quantify similarity between images. During training, the model is encouraged to assign similar images similar model outputs, and dissimilar images distant model outputs. In the absence of any prior information about the data samples, the most popular way of imposing a notion of which images are similar is to make small perturbations to an image, and then define the image and its perturbation to be more similar than the image and other images drawn from the dataset. The augmentations that form these perturbations are typically transforms such as flips, crops, rotations, blurs and color channel jittering. Popular realizations include SimCLR [8] and MoCo v2 [9]. SimCLR was previously adapted for phenotypic screening in [31] where it is applied to the BBBC021 dataset for state of the art [19] results under the NSC metric.

Direct application of contrastive learning methods such as SimCLR result in a model which describes chemically induced variation of interest but also other ways in which cells are changing irrelevant to downstream interpretation of compound effects. In this study we suggest that predicting consistency of image representations across a well is less susceptible to learning the type of variation within a well, leading to improved embeddings. To use well metadata inside a contrastive setup, we pursue a scheme where the wells are represented as non-parametric class vectors. We then optimize the model with a contrastive loss adapted to compare images with the non-parametric well representations. The well representations are improved with a simple update rule. Non-parametric representations have been successfully used in contrastive learning before in [35] where it is demonstrated that an approach of this type can be effective with over a million non-parametric vectors. Our method differs from [35] in the choice of update rule, as well as the conceptual purpose of the non-parametric vectors and some implementation details. In Section 2 we justify our choice of update rule with improved training curves.

Code is available at: https://github.com/FrancisCrickInstitute/spherical-phenotype-clustering

2 Method

A simple way to incorporate well information during training is supervised learning with wells as targets. However in datasets with appreciable numbers of plates, this soon runs in to the problem of a classifier with very large numbers of classes. Specifically, if we assume we have k images per well and N images overall, then the number of classes will be N/k. Computing a softmax for this number of classes becomes infeasible when the dataset is large [27, 35]. Furthermore, explicitly learning similarity between images makes contrastive learning well suited to the downstream task of quantifying treatment similarity, where new compounds are presented that may not neatly fall in to an existing classification bin.

The proposed alternative is to maintain a vector representation of each well, and use a loss function that operates exclusively on wells within a batch. Vector representations are non-parametric in the sense that they are not optimized by gradient updates. Instead, we provide a separate update rule to improve the well representations. We work with the NTXent loss function [33, 8]:

$$L = -\sum_i \log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{w}_i)/\tau)}{\sum_j \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{w}_j)/\tau) + \sum_{j \neq i} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)} \tag{1}$$

Where the input image has batch index i and the summations are over all images in a batch, $\mathrm{sim}(a, b)$ is the cosine similarity, τ is a temperature parameter and $\boldsymbol{z}_i$ refers to the output of a deep learning model. In SimCLR $\boldsymbol{w}_i$ denotes an augmentation of $\boldsymbol{z}_i$. For our method we instead define $\boldsymbol{w}_i$ to be the vector for the well from which $\boldsymbol{z}_i$ originated. The computation of Eq. (1) is independent of dataset size.

By using cosine distance as the metric in Eq. (1) we are implicitly learning angular relationships between vectors. As has been done in previous work [35, 15] we make the angular distribution explicit by normalizing all output of the model to unit length. This means that the output distribution lies on an n-sphere. To update the well positions we identify the output of a batch as class prototypes and move vectors towards prototypes (Figure 2A).

On a two dimensional sphere the shortest path (geodesic) between two points confined to the surface lies on a great circle. A well known fact of high dimensional geometry is that this also holds on an n-sphere. More specifically, the intersection any 2-plane passing through the origin

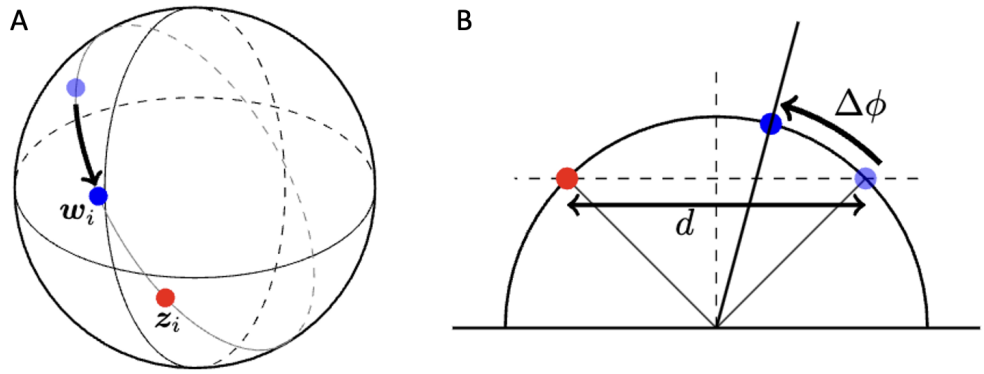

Figure 2: A) Well vectors are updated by incrementally moving toward recently computed output B) Inspecting the intersecting plane provides the form of Eq. (2).

with a unit n-sphere contains n-sphere shortest paths between points on the resulting circle. We define a 2-plane using $\boldsymbol{z}_i$ and $\boldsymbol{w}_i$. By inspecting Figure 2B it is straightforward to read off, for a specified angular velocity, the increment along the chord connecting the two points:

$$\boldsymbol{w}_i^{(new)} = \boldsymbol{w}_i + \left(\frac{d}{2} - \sqrt{1 - \frac{d^2}{4}} \tan\left(\arcsin\left(\frac{d}{2}\right) - \Delta\phi\right)\right) \frac{\boldsymbol{z}_i - \boldsymbol{w}_i}{d} \tag{2}$$

Where $d = |\boldsymbol{z}_i - \boldsymbol{w}_i|$ and $\Delta\phi$ denotes the angle a well is to be moved in a single update. After computing this value we project the well representation back out to the unit n-sphere. This update is performed directly after a gradient update to the model. Since the optimization objective in Eq. (1) encourages individual well images to be close to the corresponding well vector, the optimization objective can be seen as a form of centroid clustering. Images are averaged by well before visualizations and metric reporting, making the algorithm unsupervised (in the same way that comparing crops from the same image is an unsupervised algorithm).

We considered two alternatives for the update rule. These are a) setting the new value for the vector to the prototype value (done in [35]), b) incrementally moving along the chord connecting the two points. However we find the surface update gives better performance and convergence than the other options (see Figure 3).

Note that it is also possible to use parametric vectors instead of Eq. (2). By using Eq. (1) and thereby restricting the application of gradients to those well vectors that appear in a batch we

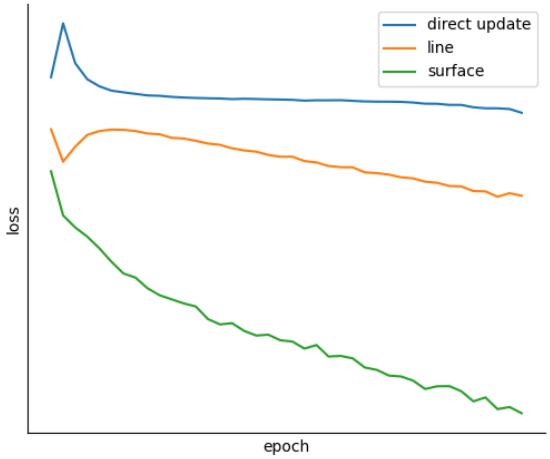

Figure 3: Constraining well vectors to move along the spherical surface avoids an undesirable 'bump' sometimes appearing in loss curves. It also converges faster and to higher numbers. Data in this image was generated training on brightfield images which shows trends particularly clearly.

have an analogous parametric formulation. However we found weak performance in experiments and instead pursued the non-parametric approach.

In the setup described above we have focused on representations of wells. An advantage of this is consistency with the common practice of examining low dimensional visualizations of well embeddings. However there is nothing in the technique that is specific to wells; any metadata labels can be chosen for use as the non-parametric representations. In Section 3.1 we also report results using treatment labels.

2.1 Batch effect correction

Previous work [1, 14, 31] has focused on postprocessing for the removal of batch effects. However, this type of postprocessing does not correct clear separation of similar or identical treatments by batch. See for example Supplementary Figure 2B in [1]. The issue is closely related to the wider machine learning challenge of data distribution shift, but is particularly acute in the phenotypic space where there are data subgroups corresponding to the date an experiment was performed, and the plate samples were prepared on.

A recent approach to this problem is found in [23]. Along the lines of previous computer vision work [21] 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). They also suggest skewed sampling during training. We do not perform the skewed sampling and implement a slightly different evaluation strategy. 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 [30] by using a BatchNorm layer with momentum 1.0, and at evaluation time performing 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 Experiments

We perform experiments on two datasets, the public BBBC021 dataset and a dataset of HaCat cells treated with JUMP compounds. Unless otherwise stated, in what follows all experiments had a training run of 100 epochs with the Adam optimizer [17] and an initial learning rate of 0.0003. A ResNet18 [13] was used as the model and the update angle $\Delta\phi$ of Section 2 is fixed at $\pi/160$ each update. Further hyperparameter details can be found in the configuration files on our GitHub repository.

3.1 BBBC021

For comparison with other approaches on a publicly available dataset we use BBBC021 [4, 25]. BBBC021 is comprised of MCF-7 cells treated with 113 small molecules at eight different concentrations. Most work has focused on a curated subset of 103 treatments annotated with MoA. This subset is chosen based on visibility of phenotypes and prior literature. Our benchmarking is performed on this portion of the data.

We perform similar preprocessing to that in [1] including using the illumination correction functions [32], log transforming data and making use of the single cell crops defined in [24]. But

Method	Type	NSC	NSCB
Ljosa et al. [24]	CellProfiler + Factor Analysis	94%	77%
Singh et al. [32]	CellProfiler	90%	85%
Lafarge et al. [20]	Autoencoder	92.9%	82.2%
Janssens et al. [14]	DeepCluster	97%	85%
Perakis et al. [31]	Contrastive Learning (SimCLR)	97%	94%
Ando et al. [1]	Transfer Learning	96%	95%
Caicedo et al. [3]	Treatment Classifier (DeepProfiler)	97%	86%
Ours (wells, all data)	Contrastive Learning (SPC)	98.2%	95.4%
Ours (wells, holdout)	Contrastive Learning (SPC)	96.0%	93.0%
Ours (treatment, all data)	Contrastive Learning (SPC)	99.4%	98.1%
Ours (treatment, holdout)	Contrastive Learning (SPC)	97.0%	95.5%

Table 1: NSC/NSCB scores. We run training 10 times and report the mean across runs.

we rescale channels independently and do not perform postprocessing, instead using the batch correction described in Section 2.1.

The evaluation scheme is as follows [1, 14, 31, 24, 32, 3]. After training a model all embedding vectors originating from a well are mean averaged to create a single well representation, then for each treatment, the median embedding is computed resulting in one vector per treatment. After this two MoA classification metrics are calculated. The first is the NSC (not-same-compound score), wherein each treatment vector is compared to the closest embedding not labelled with the same compound, and an accuracy is reported under the criterion of the vectors having the same MoA label. The NSCB (not-same-compound-or-batch) is the same except the neighbour embedding cannot be from the same batch.

The highest performing unsupervised methods use all data during training and evaluation, and select the epoch with highest NSC score for reporting [14, 31]. We report metrics under this scheme, and also on holdout data in a conventional train/test split. In the second case we use 60% of the cells for training and model selection, and the other 40% as holdout data. Both splits contain samples from all treatments and we compute the holdout NSC/NSCB using only holdout samples. A previous section mentioned that our technique is agnostic to exactly which metadata

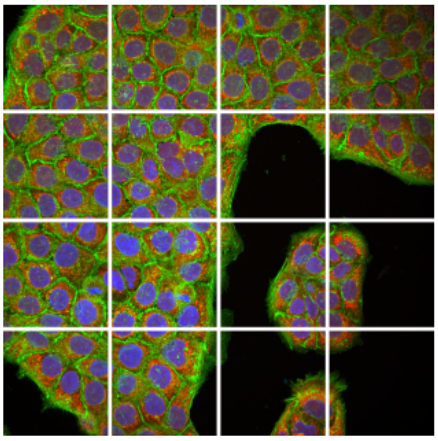

Figure 4: Raw 1080x1080 images are resized to 512x512 and then tiled.

is used to train the algorithm. In this instance metrics are reported after all replicates for a given treatment have been aggregated so we also report performance training with treatment labels. Two methods we have included for comparison use treatment labels during training [3, 14].

We report results in Table 1. Our method gives improved NSC and NSCB scores when fitting all data, and it also performs well on data not seen during training.

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.

3.2 HaCat JUMP compound dataset

Our new cell painting dataset is composed of HaCaTs, a spontaneously immortalised human keratinocyte lineage. We chose to screen a GSK version of the JUMP-MoA library, initially developed by the Joint Undertaking for Morphological Profiling (JUMP)-Cell Painting Consortium, which aims to generate a large publicly available cell painting dataset [7]. The JUMP-MoA library comprises 90 compounds in quadruplicate, covering 47 distinct MoA classes [10], and thus serves as an ideal compound set to elicit a broad range of cellular phenotypes. For the cell painting assay, we followed the cell painting assay protocol [10, 2], except for optimisations of dye

Method	KNN
SimCLR (setup used in [31])	42%
CellProfiler	34%
SPC	**56%**

Table 2: Predicting MoA on JUMP HaCat dataset with KNN where k=1

concentrations to achieve optimal staining in our cell line. Screening was performed at triplicate concentrations (0.1, 1.0 and 10 uM), adding MitoTracker 72h after cell seeding, followed by cell fixation, permeabilisation, and staining for the following subcellular components: DNA (Hoechst 33342), endoplasmic reticulum (concanavalin-A 488), RNA and nucleoli (Syto-14), golgi and plasma membrane (wheat germ agglutinin Alexa-555), and F-actin (phalloidin Alexa-568). Images were acquired in four channels at 40X magnification using an Opera Phenix Plus high content microscope, and max projected.

To preprocess this dataset we resized each image to 512x512 pixels and generated 16 crops under a 4x4 tiling scheme [1] (see Figure 4). 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 clipping thresholds we use are the 0.01 and 99.9 percentiles as in [16].

Our evaluation method is a simple 1-nearest-neighbour scheme. After training we do a forward pass on all tiles. We then compute the mean of the embedding vectors inside a single well before projecting back out to unit radius. For each well we compare its MoA with that of its closest neighbour. We report the accuracy over all wells in Table 2 using the model from the final training epoch.

A UMAP of well embeddings reveals qualitative clustering aligned with MoA and treatment

[1] 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.

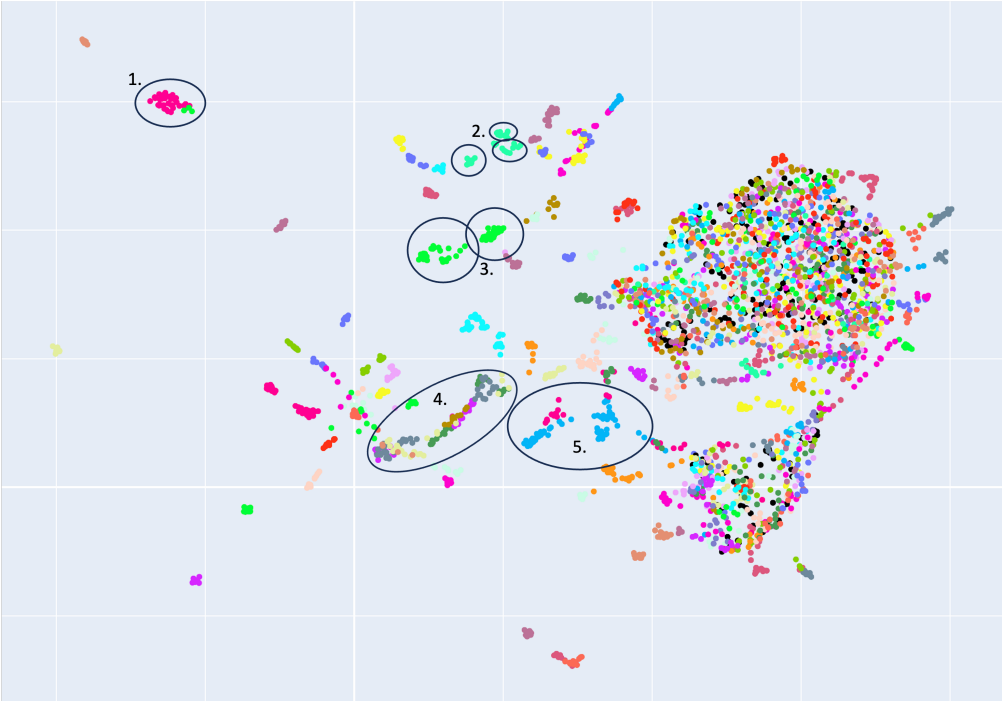

Figure 5: UMAP visualization with several representative clusters highlighted. 1) Niacinamide phosphoribosyltransferase inhibitor 2) Palbociclib mini-clusters corresponding to three concentrations 3) Aurora kinase inhibitors 4) MAPK/ERK pathway, increasing concentration moving left 5) mTOR inhibitors locally structured by compound, concentration. Plot coloured by MoA.

labels (Figure 5). A large cluster is visible where treatments have not been distinguished from controls, as well as a good number of smaller groupings related to MoA and treatment. In some cases such as MTOR inhibitor, multiple treatments are found within one cluster, but locally structured such that concentrations can be distinguished. In other cases concentrations are separated in to different clusters.

One reason for pursuing an unsupervised approach was the hope we would see phenotypic associations not known directly from MoA labels. Several 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). We might see the alignment of these compounds as a limitation of the technique. However an alternative explanation is offered by all three lying on the MAPK/ERK signalling pathway [29]. Inspection of the images gives marginal visual confirmation (Figure 6). However the circumstance is clearer if we apply TGF-

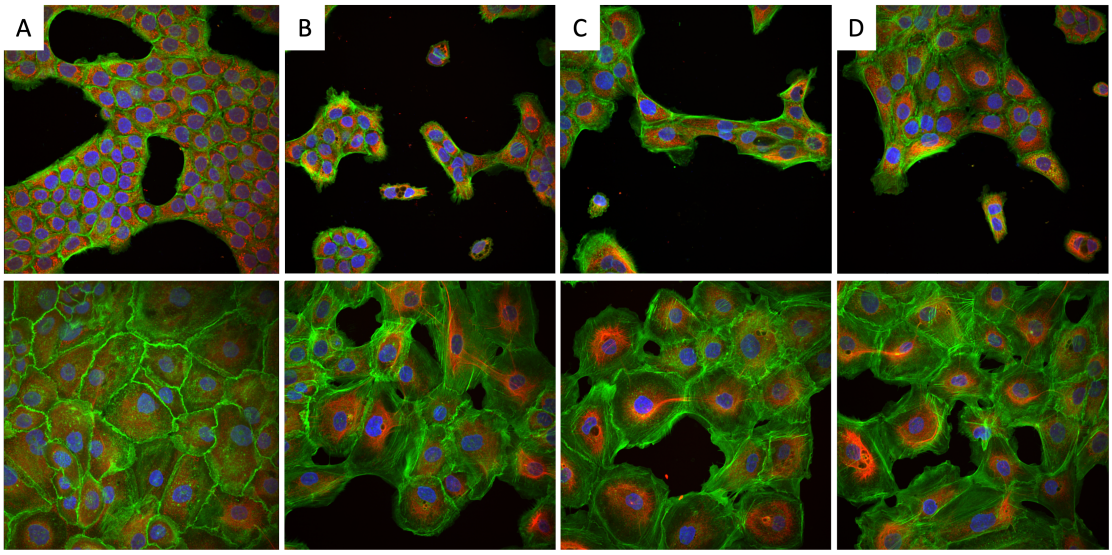

Figure 6: First row: no TGF-beta stimulation. Second row: TGF-beta stimulated. A) negative control B) neratinib@0.1uM C) selumetinib@0.1uM D) PD-198306@1uM

beta stimulation to the cells. Under these conditions the cells become characterised by activity 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.

4 Conclusion

While some work has abandoned the use of metadata in favour of learning objectives based on augmentations or masking [31, 18], it is a notable feature of HCS that this metadata means some level of semantic meaning is always attached to images. We have demonstrated recovering the use of this semantic information outperforms alternatives on datasets on the order of thousands of wells. Our approach avoids loss function scaling issues of supervised learning, but it remains to demonstrate empirically it is performant on large datasets. A recent trend is the exploration of vision transformer models [16, 11, 18]. The quantity of data cited for vision transformers to outperform CNNs [12] goes beyond datasets included in this work, however we note that our method is model agnostic so is at least in principle compatible with vision transformers. Future investigation will explore these questions of scalability.

References

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

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

[3] 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.

[4] 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.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[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.

[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.

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

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

[10] 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.

[11] 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.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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 are scalable learners of cellular morphology. *arXiv preprint arXiv:2309.16064*, 2023.

[19] 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.

[20] 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.

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

[22] 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*, volume 35, pages 8547–8555, 2021.

[23] 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.

[24] 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.

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

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

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

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.