Learning unsupervised feature representations for single cell microscopy images with paired cell inpainting

Alex X Lu[1], Oren Z Kraus[2], Sam Cooper[2], and Alan M Moses[1,3,4]

[1]Department of Computer Science, University of Toronto, Toronto, Canada
[2]Phenomic AI, Toronto, Canada
[3]Department of Cell and Systems Biology, University of Toronto, Toronto, Canada
[4]Center for Analysis of Genome Evolution and Function, University of Toronto, Toronto, Canada

Abstract

We introduce an unsupervised deep learning method that learns features for representing single cells in fluorescence microscopy images. Our method exploits the powerful capacity of convolutional neural networks in learning image features, by training networks on a simple self-supervised task that leverages the structure of microscopy images: given one cell from an image, the network is asked to predict the appearance of a second different cell from the same image. We demonstrate that the features learned through our training task surpass other unsupervised feature sets in discriminating phenotypes for both yeast and human cells. Moreover, we demonstrate that our features are useful for exploratory biological analysis, by capturing fine-grained phenotypes in a proteome-wide cluster analysis of human proteins, and by showing that our features can represent rare and complex phenotypes.

Introduction

Feature representations of cells within microscopy images are critical for automated image analysis applications such as assessing the effects of pharmaceuticals on cancer cells (1), or detecting changes in subcellular protein localization in response to environmental perturbation (2,3). Most work to date has relied on low-level intensity, texture, or shape features extracted through classic computer vision pipelines (4–7). In contrast, deep convolutional neural networks (CNNs) have shown superior performance in classifying protein localization in yeast cells (8), cell-cycle stages in imaging flow cytometry (9), and morphological phenotypes in high-content screening images (10,11). Since CNNs automatically learn feature representations optimized to their classification tasks directly from pixel-level data (12), their features may be more sensitive to biological properties within the images relative to classic approaches, as evidenced by their good performance when transferred to other tasks or datasets (8,13,14).

A key limitation in learning CNN features through classification tasks is that it relies on manually annotated datasets, which are expensive and time-consuming to generate. Towards the goal of developing high-quality CNN features without annotated data, self-supervised learning methods have been proposed. These methods train CNNs using information that is automatically available or synthesized from the input images. In the context of natural images, Pathak et al. (2016) train CNNs to predict the appearance of withheld image patches based upon its context (15), Jenni and Favaro (2018) predict the presence and location of synthetic artifacts in images (16), and Gidaris et al. (2018) predict geometric rotations applied to input images (17). These methods rely on the premise that to succeed at the self-supervised task, the CNN needs to develop a strong internal representation of the objects and patterns in the images. When transferred to vision tasks such as classification, segmentation, or detection, features developed by self-supervised methods have shown state-of-the-art results compared to other unsupervised methods (16–20).

Here, we present a novel self-supervised learning method designed for cellular microscopy images. Our approach leverages the typical structure of microscopy images: in many cases, an image will contain multiple instances of cells, and these cells will usually exhibit similar phenotypes as they have been

grown under the same experimental condition. We use high-content microscopy images where cells have been tagged with markers visualizing one or more structures in the cell (e.g. the cytosol, nucleus, or cytoskeleton), and where proteins of interest have been fluorescently tagged to visualize their localization (with a different protein tagged in each image set). Then, given one cell from an image, and the structural markers for a second different cell from the same image, our CNN is trained to predict the appearance of the protein of interest in the second cell, a task that we term "paired cell inpainting" (Figure 1). Our task teaches the CNN to identify protein localization in the first (or "source") cell to reconstruct a similar protein localization in the second (or "target") cell in a way that adapts to single cell variability. In Figure 1, the protein is localized to the nucleoli of human cells – the network must recognize the localization of the protein in the source cell, but also transfer it to the equivalent structures in the target cell, in spite of differences in the morphology of the nucleus between the two cells.

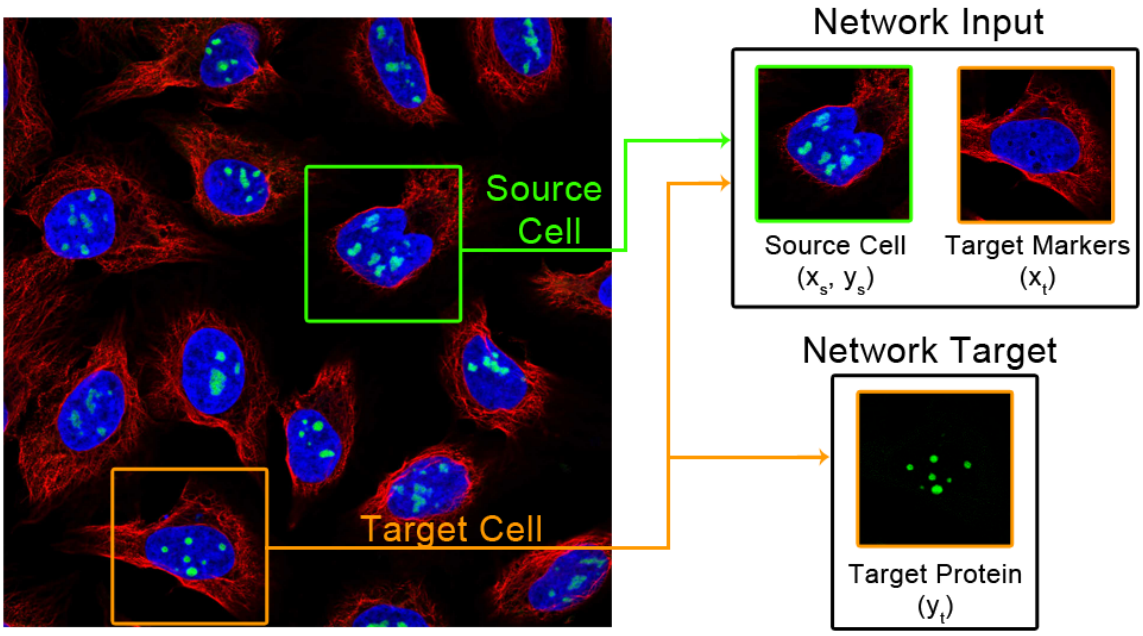


Figure 1. An overview of the inputs and targets to the network in paired cell inpainting. Images shown are of human cells, with the nucleus colored blue, microtubules colored red, and a protein localized to the nucleoli colored green. We crop a source cell (green border) and a target cell (orange border) from the same image. Then, given all channels for the source cell, and the structural markers for the target cell (in this dataset, the nucleus and the microtubule channels), the network is trained to predict the appearance of the protein channel in the target cell.

We demonstrate that paired cell inpainting is a flexible learning task, and can be applied to images of both human and yeast cells; the features learned through our method exhibit improvements in discriminating protein subcellular localization classes in both organisms compared to other unsupervised methods. We demonstrate that our learned features are useful for biological applications by performing the first proteome-wide cluster analysis of protein localization in human cells, which captures fine-grained enrichments in localization, and by smaller-scale single cell analyses, which distinguish spatially variable single cells and multiply-localized single cells.

Methodology

Paired Cell Inpainting

We would like to learn a representation for single cells in a collection of microscopy images, I. We define each image i as a collection of single cells, $i = \{c_{i,1} \dots c_{i,n}\}$. We note that the only constraint on i is that its single cells C_i must be considered similar to each other, so i does not need to be strictly defined as a single digital image so long as this is satisfied; in our experiments, we consider an "image" i to be all fields of view corresponding to an experimental well.

We define single cells to be image patches, so $C \in \mathbb{R}^{H \times W \times Z}$, where Z are the channels. We split the images by channel into $C = (X, Y)$, where $X \in \mathbb{R}^{H \times W \times Z_1}$, $Y \in \mathbb{R}^{H \times W \times Z_2}$, and $Z_1, Z_2 \subseteq Z$. For this work, we assign to Z_1 channels corresponding to structural markers, or fluorescent tags designed to visualize structural components of the cell, where all cells in the collection of images have been labeled with the tag. We assign to Z_2 channels corresponding to proteins, or channels where the tagged biomolecule will vary from image to image.

We define a source cell c_s, which is associated with a target cell c_t satisfying constraints that both cells are from the same image, $c_s \in i_s, c_t \in i_t, i_s = i_t$, and $c_s \neq c_t$. Then, our goal is to train a neural network that will solve the prediction problem $\widehat{y_t} = f(x_s, y_s, x_t) \forall c_s, c_t \in I$.

For this work, we train the network on the prediction problem by minimizing a standard pixel-wise mean-squared error loss between the predicted target protein $\widehat{y_t}$ and the actual target protein y_t:

$$L(\widehat{y_t}, y_t) = \frac{1}{h \cdot w} \sum_{h,w} (\widehat{y_t}_{h,w} - y_{t_{h,w}})^2$$

One limitation in our training task is that some protein localizations are not fully deterministic in respect to the structure of the cell, and are therefore challenging to predict given the inputs we define. In these cases, we observe that the network produces smoothed predictions that we hypothesize are an averaged guess of the localization. We show an example in Figure 2; while the source protein is localized to the nucleus in a punctate pattern, the target protein is predicted as a smooth distribution throughout the nucleoplasm. However, as the main goal of our training task is to develop the internal feature representation of the CNN, we accept that our network will sometimes output imprecise images depending on the stochasticity of the source protein.

Architecture

As the goal of our training task is to obtain a CNN that can encode single cell image patches into feature representations, we construct independent encoders for (x_s, y_s) and for x_t, which we call the "source cell encoder" and the "target marker encoder", respectively. This architecture allows us to isolate the source cell encoder after training with our self-supervised task, and encode any single cell image patch independently without also having to input target cell markers. We show a summary of our architecture in Figure 2.

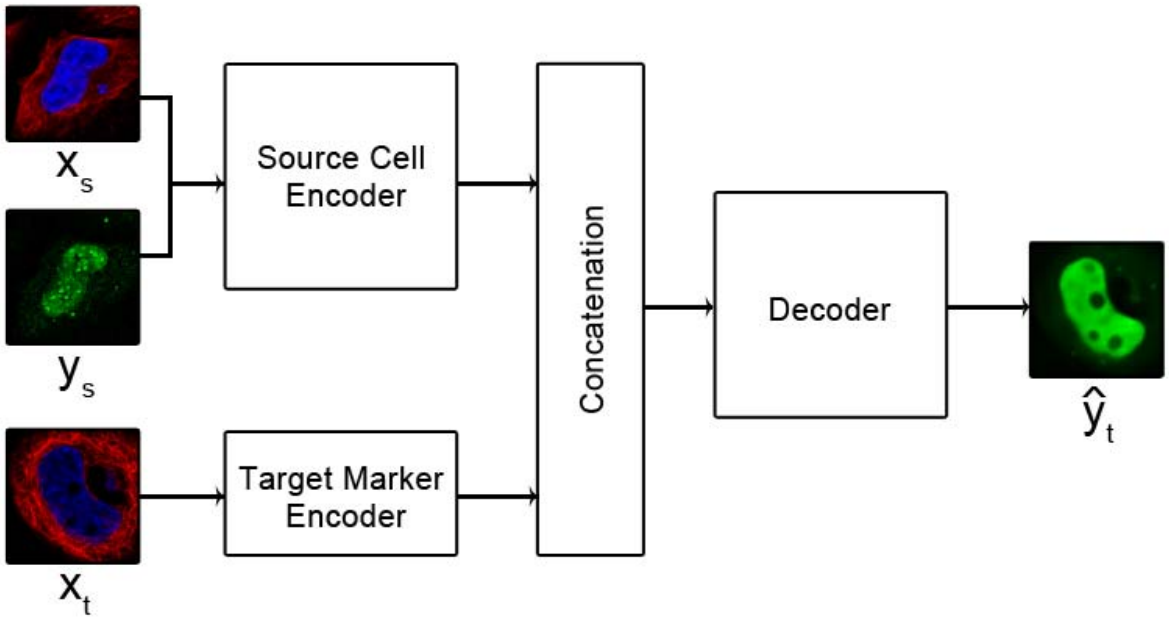


Figure 2. Our proposed architecture. Our architecture consists of the source cell encoder and the target marker encoder. The final layers of both encoders are concatenated, and fed into a decoder that outputs the prediction of the target protein $\hat{y}_t$. We show a real example of a prediction from our trained human cell model given the input image patches in this schematic. The image patches are of human cells, with the nucleus colored blue, microtubules colored red, and the protein colored green. In this case, the source protein is localized to the nucleus in a punctuate pattern. Micrographs have been rescaled by stretching the contrast to the maximum image range.

Following other work in self-supervised learning (15,17,18), we use an AlexNet architecture for the source cell encoder, although we set all kernel sizes to 3 due to the smaller sizes of our image patches, and we add batch normalization after each convolutional layer. As there can sometimes be artifacts in the target markers that indicate the target protein in the form of bleed-through between fluorescent channels (21), we attempt to limit the information extracted from the target markers by using a smaller number of filters and fewer convolutional layers in the architecture of the target marker encoder; we use three convolutional layers, with 16, 32, and 32 filters, respectively. Finally, for the decoder, we reverse the AlexNet architecture.

Sampling Pairs

Because our inputs consist of pairs of cells, the number of possible combinations is large, as each cell may be paired with one of many other cells. This property is analogous to data augmentation, increasing the number of unique training inputs to our network. To sample inputs, we employ a simple random sampling scheme. For each epoch, we iterate through every single cell in our training dataset, set it as the source cell, and draw with uniform probability a target cell from the set of all valid target cells for that cell.

Our method relies on the assumption that all single cells for an image are phenotypically similar. In a fraction of images in a dataset, this assumption may be violated. For example, in the datasets used in this work, some proteins exhibit significant single cell variability in their protein abundance or localization

(22,23). If these proteins are known a priori, they can be removed from the training dataset. If not, we introduce additional constraints in our sampling to encourage pairings of more similar cells.

If we don't know which images contain single cell variability, we will use a k-nearest neighbor sampling scheme instead. We associate a feature vector m_c, which consists of basic image or object-level measurements of protein expression, with each cell c. In our experiments, we set m to be a single measurement of the average intensity of protein expression in the cell. For a given source cell c_s, we quantitate its similarity to a target cell c_t using the Euclidean distance between their feature vectors:

$$D(c_s, c_t) = \sqrt{\sum_{i=1}^{n} (m_{c_s,i} - m_{c_t,i})^2}$$

Then, when sampling a target cell for a given source cell, we draw with uniform probability a target cell from the set of valid target cells with the k smallest $D(c_s, c_t)$ values only. In other words, we only pair cells with the k most similar cells in the image, instead of all cells in the image.

Datasets, Preprocessing, and Training

For our experiments, we train our models using image datasets where proteins have been systematically visualized with fluorescence microscopy. To render these datasets appropriate for our model, we extract single cell crops from the images.

For our models for yeast cells, we use the WT2 dataset from the CYCLoPS database (24). This collection expresses a cytosolic red fluorescent protein (RFP) in all cells, and tags proteins of interest with green fluorescent protein (GFP). We use the RFP channel as the structural marker, and the GFP channel as the protein. To extract single cell crops from the images in this dataset, we use a single cell segmentation program designed for yeast cells (6), and extract a 64x64 pixel crop around the identified cell centers. As the single cell segmentation program is sensitive to artifacts in the image, we filter cells that have an area of less than 800 pixels. As a large proportion of the yeast proteome is expressed at low abundance indistinguishable from autofluorescence (2), we also filter cells that have a mean GFP intensity value of less than 50.0 A.U. in order to increase the proportion of training data with recognizable protein localization patterns. After these filters, we discard any images with fewer than 20 cells. These preprocessing operations result in a total of 333,130 single cell image patches over 1,734 images (where each image is 4 fields of view).

For our yeast model, we train a model for 100 epochs on our WT2 yeast cell dataset, using an Adam optimizer with an initial learning rate of 1e-4. We drop the learning rate to 1e-5 at 30 epochs, and to 1e-6 at 70 epochs. We sample target cells using a $k$nn sampling scheme, with $k = 20$.

For our models on human cells, we use images from the Human Protein Atlas (25). We downloaded images for a total of 12,068 proteins. Each protein may have multiple experiments, which image different cell line and antibody combinations. We consider an image to be of a protein for the same cell line and antibody combination; accordingly, we have 41,517 images (where each image is usually 2 fields of view). We downloaded 3 channels for these images. Two visualize the nuclei and microtubules, which we use as the structural marker channels. The third channel is an antibody for the protein of interest, which we use as the protein channel. To extract single cell crops from this image, we binarize the nuclear channel with an Otsu filter and find nuclei by labeling connected components as objects using the scikit-image package (26). We filter any objects with an area of less than 400 pixels, and extract a 512x512 pixel crop around the center of mass of remaining objects. To reduce training time, we rescale the size of

each crop to 64x64 pixels. As the human protein atlas contains information about which proteins contain single cell variability (25), we remove these proteins from the dataset. Finally, we remove any images with fewer than 5 cells, leaving a total of 611,981 single cell image patches over 34,264 images.

For our human models, we trained a model for 30 epochs using an Adam optimizer with an initial learning rate of 1e-4. We sample target cells randomly, as proteins with single-cell variability have been filtered from this training dataset.

To reduce bleed-through, we clip the upper intensity ranges of the RFP channel at 20.0 A.U in yeast images, and clip the lower intensity ranges of the nucleus and microtubule channels at 13.0 A.U in the human images. Finally, we normalize the image intensity of each single cell image patch to be in the range [0, 1]. During training, we apply random horizontal and vertical flips to source and target cells independently as data augmentation.

Experimental Results

Discriminating Protein Localization in Yeast Cells

To evaluate the capacity of our model to learn a feature representation that discriminates between protein subcellular localization classes in budding yeast cells, we use a set of 30,889 single cell image patches manually assigned to 17 different protein subcellular localization classes by Chong *et al*. (2). We used the source cell decoder to extract features from each cell. We use a leave-one-out *k*NN classifier to evaluate the performance of classifiers built on the features of each layer in the source cell decoder in Table 1; we set $k = 11$, as this resulted in the best overall performance in all feature sets in our yeast cell benchmarks. Overall, we observe the best performance from Conv3 and Conv4.

Layer	Overall Classification Accuracy
Conv1	37.18
Conv2	63.46
Conv3	**75.37**
Conv4	**73.90**
Conv5	53.83

Table 1. Classification accuracies using various layers in our CNN, for single yeast cell localization classes using a leave-one-out *k*NN classifier ($k = 11$). For each layer, we take the maximum activation of each feature map. We report the overall accuracy as the balanced accuracy of all classes.

Next, we benchmarked the performance of our features against other unsupervised feature sets (Figure 3A). We compared our features with CellProfiler features (5) (previously extracted by Chong *et al*. to train their classifier (2)), and with handcrafted features measuring yeast protein localization from Handfield *et al*. (6). We also compared our features with those obtained from transferring a VGG16 model pretrained on ImageNet (27), and an autoencoder using our source cell encoder and decoder architecture. As with before, we use a leave-one-out *k*NN classifier with $k = 11$. The CellProfiler, and VGG16 features perform at 68.15% and 69.33% accuracy respectively on our evaluation set, while the Handfield features are worse at 62.04%. Autoencoder features achieve 42.50% accuracy. Our features perform better than any of the other features benchmarked, achieving 75.37%.

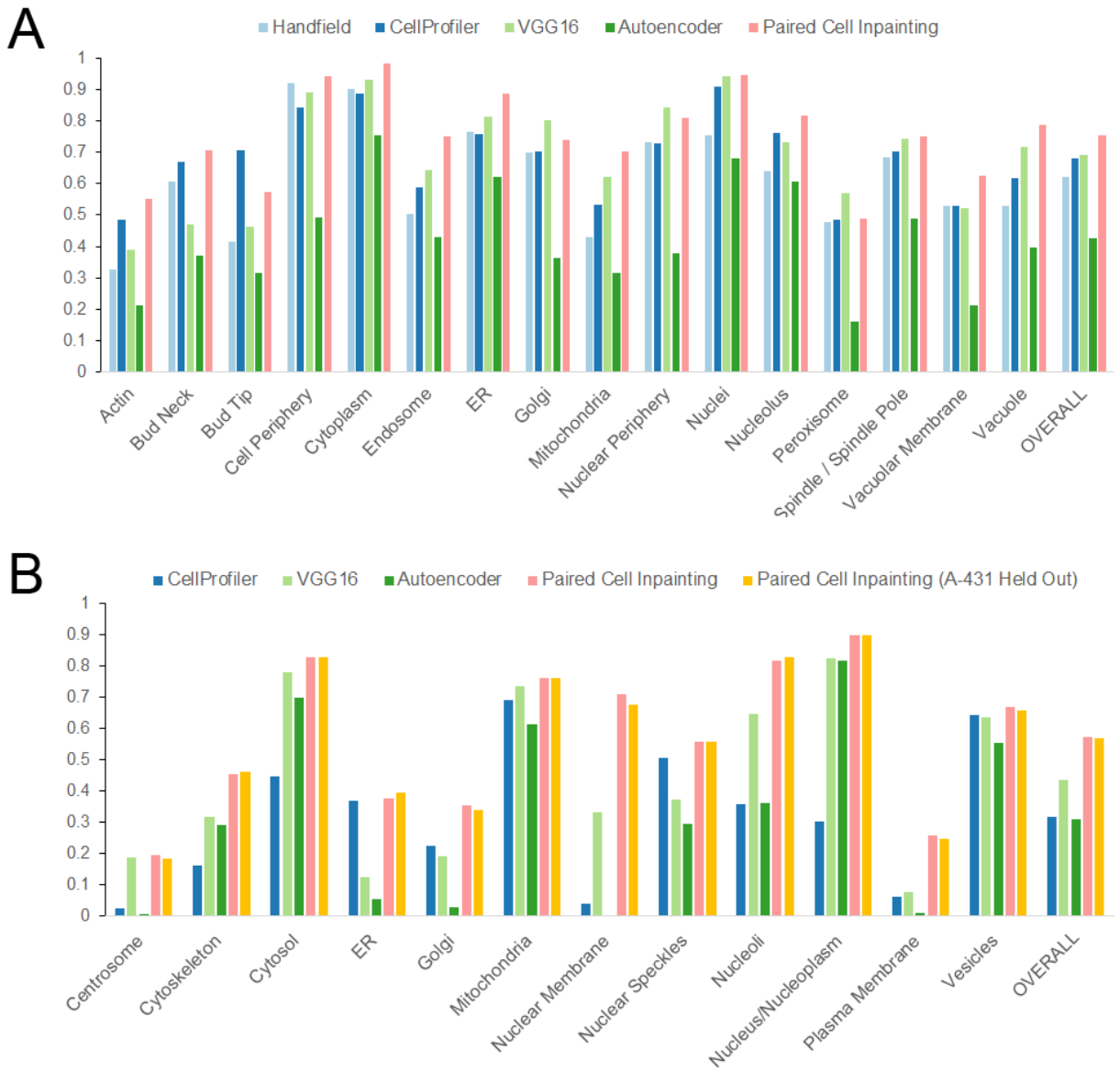


Figure 3. Overall and class-by-class performance benchmark for yeast (A) and human A-431 (B) single cell protein localization classes using unsupervised feature sets using a leave-one-out *k*NN classifier ($k = 11$ for yeast cells; $k = 25$ for human cells). For VGG16, Autoencoder, and Paired Cell Inpainting, we take the maximum activation of each feature map, and report the top-performing layer. We report the overall accuracy as the balanced accuracy of all classes. For human classifiers, we hold out all single cells from the same image when predicting labels for cells. For human data, we show accuracy for two models trained using paired cell inpainting: one that has been trained on all data, and another that has been trained with all A-431 images held out.

Discriminating Protein Localization in Human Cells

Next, we assessed our model in learning discriminative features for protein subcellular localization classes in human cells. To construct an evaluation set, we use single cells in the A-431 cell line, using the subcellular localization annotations contained in the human protein atlas (25). We filter our dataset to contain only proteins annotated to localize to a single compartment without single cell variability, and

assign the annotation to all cells in the image. Because this dataset is heavily class-imbalanced (for example, there are only 40 images of centrosome proteins, but 1,065 of nuclear proteins), we randomly downsample images in overrepresented classes such that each class only has a maximum of 300 images. Overall, we obtained an evaluation set comprising of 1,701 images spanning 12 localization classes.

As in our yeast cell evaluation, we compare our features to features extracted on single-cell segmentations by CellProfiler, and image patches by a pretrained VGG16 model. One complication of the human cell evaluation set is that multiple cells can originate from the same image, so a classifier could perform well by overfitting to image-specific illumination effects or experimental variation. To control for this effect, we show classification accuracies using a *k*nn classifier ($k = 25$) where we hold out all single cells from the same image (Figure 3B). Overall, we achieve 57.34% accuracy, while CellProfiler, VGG16, and autoencoder features achieve 31.83%, 43.52%, and 30.97%, respectively.

Next, we asked how well our models generalize to unseen data. Unlike supervised methods, our method can be trained on unlabelled data. However, there are some cases where having a generalizable model may be desirable. First, the model may be time-consuming to train on very large datasets, so retraining as new datasets are generated may pose a bottleneck. Second, not all images are appropriate for training the model, such as in the case of proteins with single-cell variability or images where there is only one cell, but it would be useful to represent the single cells in these images still.

To assess the performance of our model on unseen data, we retrained our human cell model, but with all images of A-431 cells held out, for a total of 5,674 images held out. We then evaluated classification performance with this model, as we did before. Remarkably, we still achieve 56.92% accuracy on the evaluation set (Figure 3B), a drop of only 0.51% versus training on all human single cell data. These results suggest that our model is not overfitting to cell lines, and learns more general features.

Biological Utility of Paired Cell Inpainting Features

Having obtained single cell features, we next asked about the utility of these features in real cases of biological exploratory analysis.

First, we performed a global unsupervised analysis of protein localization in human cells (Figure 4). We averaged the features of all single cells for each protein (pooling together all single cells from different cell lines and antibodies), and clustered the averaged features using hierarchical agglomerative clustering (Figure 5). Similar clustering experiments have been performed for hand-crafted features of yeast proteins, and have found clusters that not only recapitulate previously-defined subcellular localization patterns, but discover protein localization at a higher resolution than manual assignment (6). While hierarchical clustering has previously been applied to a subset of human proteins in a single cell line to identify misannotated proteins (28), our analysis is, to our knowledge, the first to encompass all proteins in the Human Protein Atlas.

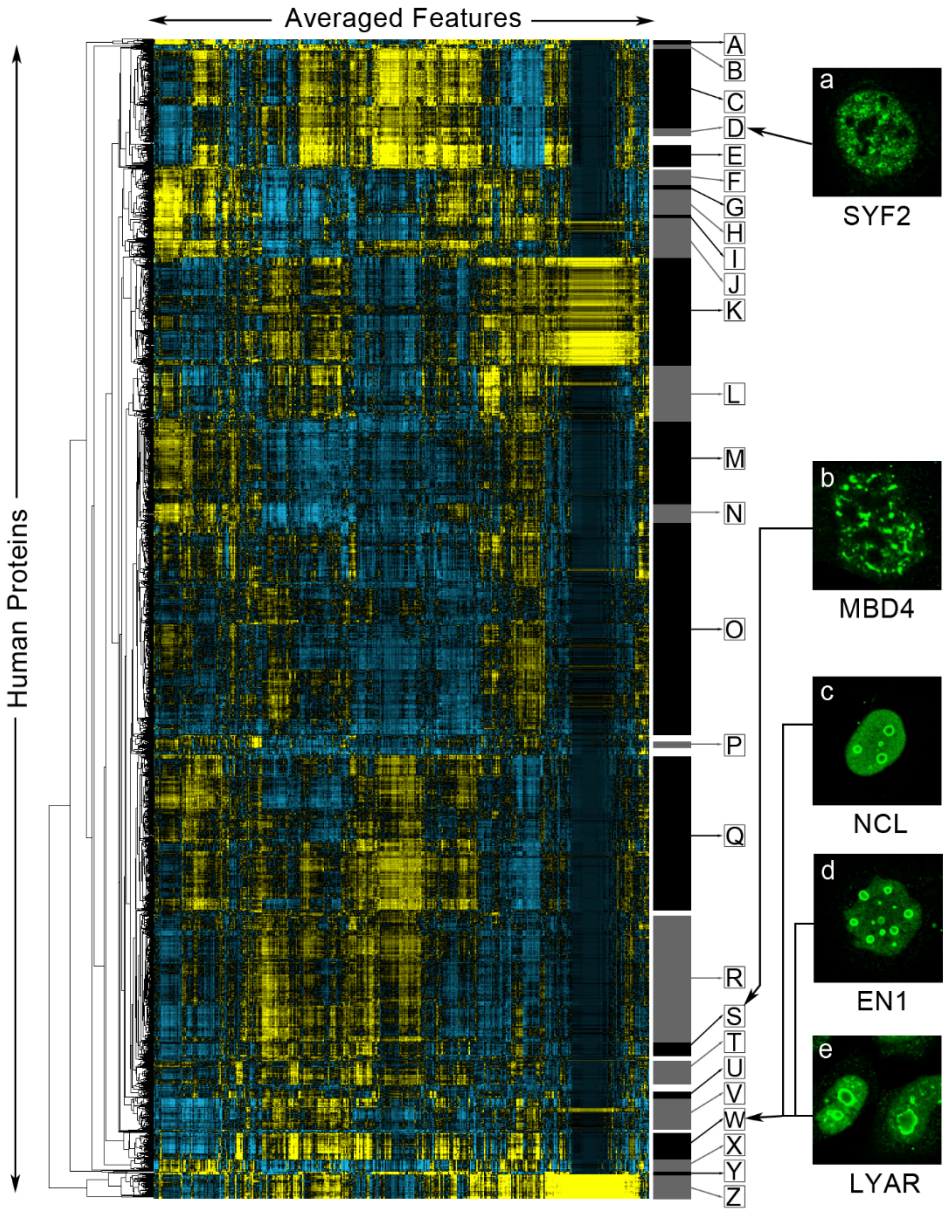


Figure 4. Averaged paired cell inpainting features for proteins in the Human Protein Atlas, ordered using maximum likelihood agglomerative hierarchical clustering. We visualize features as a heat map, where positive values are colored yellow and negative values are colored blue, with the intensity of the color corresponding to magnitude. Columns in this heat map are features, while rows are proteins. Features have been mean-centered and normalized. We manually select clusters (grey and black bars on the right) for enrichment analysis, presented in Table 2. On the right panels, we show representative examples of the protein channel (green) of cells from proteins (as labeled) in the clusters indicated by arrows from the images.

Cluster	GO Annotation	Enrichment (FDR Adjusted)	Number of Proteins
A	Extracellular Exosome	1.7E-5	21/47

B	Nuclear Membrane	1.83E-9	12/32
C	Nucleoplasm	3.74E-88	421/709
D	Nuclear Speck	3.58E-39	40/73
	Spliceosomal Complex	1.23E-21	22/73
E	Nucleoplasm	2.87E-32	123/179
	Chromosomal Part	1.35E-22	56/179
F	Cytosol	4.3E-8	79/126
G	Endoplasmic Reticulum	1.18E-23	32/54
H	Cytosol	3.16E-28	181/264
I	Ribosomal Subunit	3.45E-25	19/36
J	Cytosol	1.13E-13	186/334
K	Membrane Part	6.26E-18	420/1001
L	Mitochondrion	1.28E-72	180/508
M	Cytoplasmic Part	2.75E-18	567/791
N	Plasma Membrane	8.81E-10	70/163
O	Intrinsic Component of Membrane	4.87E-25	530/1745
P	Adherens Junction	1.18E-8	18/73
	Cell Junction	2.24E-8	24/73
Q	Nucleoplasm	1.35E-60	634/1456
	Cytosol	1.38E-11	620/1456
R	Nucleoplasm	3.08E-35	501/1244
S	Nuclear Speck	1.45E-9	21/98
T	Nucleolus	2.56E-42	87/215
U	Nuclear Membrane	5.59E-17	22/71
V	Nuclear Body	1.34E-4	40/276
W	Nucleolus	1.29E-62	105/228
	Preribosome	3.03E-31	30/228
	Fibrillar Center	1.71E-8	18/228
X	Microtubule Cytoskeleton	1.07E-12	18/114
Y	Nucleosome	4.63E-20	10/16
Z	Membrane	2.5E-4	126/228

Table 2. Gene ontology enrichments for manually identified clusters in Figure 4. For each cluster, we report selected GO annotation terms in the component ontology for *Homo sapiens*, their enrichment (adjusted with the false discovery rate), and the number of proteins with the enrichment versus the total number of proteins in the cluster. We used the GOrilla tool (29) to identify enrichments.

We find that proteins in known localization classes cluster together (Table 2). We identify clusters enriched for about 14 of the original 30 subcellular localization classes annotated by the Human Protein Atlas (30), with the caveat that not all classes are present in the GO annotation or exhibit good penetrance in cells in the images. In addition to these known classes, we identify small clusters of protein complexes or functional protein classes, including the nucleosome (cluster Y), ribosomal subunits (cluster I), and the preribosome (cluster W). These localizations are not typically identified in manual assessment, suggesting that our features may identify biological properties that are challenging to confirm by human eye.

To assess the ability of our clustering to finding rare localization patterns, we asked if we could identify proteins localized to the nucleolar rim. The human protein atlas gives an example of one protein, MKI67, that localizes to this pattern (30), but otherwise does not document nucleolar rim proteins systematically. We found MK167 in Cluster W; of the 10 proteins closest to MK167 in our hierarchical clustering, we

qualitatively identify at least 3 other proteins with cells that exhibit a nucleolar rim pattern, NCL, EN1, and LYAR (Figure 4c-e). NCL has been found to localize to the nucleolar rim through immunoelectron microscopy in mouse embryos (31), but to our knowledge, the nucleolar rim localization of EN1 and LYAR has not previously been reported.

Finally, as we observed at least two distinct clusters of nuclear speck proteins (clusters D and S), we decided to evaluate images for proteins in these clusters to determine if there is any distinction in their phenotype. We find that proteins in cluster S are characterized by a more globular, clumped pattern (Figure 4b) than proteins in cluster S, which are distributed to the nucleus more evenly (Figure 4a). Interestingly, while we find a strong enrichment for the spliceosomal complex in cluster D, which is consistent with the knowledge that splicing factors are stored inside nuclear specks (32), we find no enrichment in cluster S. These results suggest that nuclear specks may be further distinguished into more subclasses than is currently known, with different functional roles.

Next, we asked how well our features could represent single cells with multiply-localized proteins. Multiply-localized proteins have traditionally posed a challenge for methods evaluating protein localization. Most supervised methods hold out multiply-localized proteins from their training and evaluation sets (2,8,11,33), focusing on only proteins localized to a single subcellular compartment. While generative models have been proposed for modeling these proteins (34,35), they generally rely on elaborate user-defined models of the conditional dependencies between different compartments.

Using annotations in the Human Protein Atlas, we randomly selected 50 proteins annotated to localize to the cytosol and the nucleoplasm, 20 proteins to the cytosol only, and 20 proteins to the nucleus only, and produced a t-SNE scatterplot to visualize single cells stained for these proteins (Figure 5A). We observe a continuous range of cytosol and nucleoplasm-localized single cells between the cytosol only and nucleoplasm only single cells. Single cells closer to the nucleoplasm-only cluster exhibit a stronger nucleoplasm localization (Figure 5A-2), while single cells closer to the cytosol-only cluster exhibit a stronger cytosol localization (Figure 5A-4). In some cases, single cells annotated to localize to the nucleoplasm only were clustered with other cytosol and nucleoplasm-localized single cells; in most cases, we observed that these single cells had some cytoplasmic localization. For example, UBOX5 is annotated to localize to the nucleoplasm only, but we observed some weak cytosolic expression for this protein (Figure 5A-2). Overall, these results suggest that our learned features can quantify multiply-localized proteins.

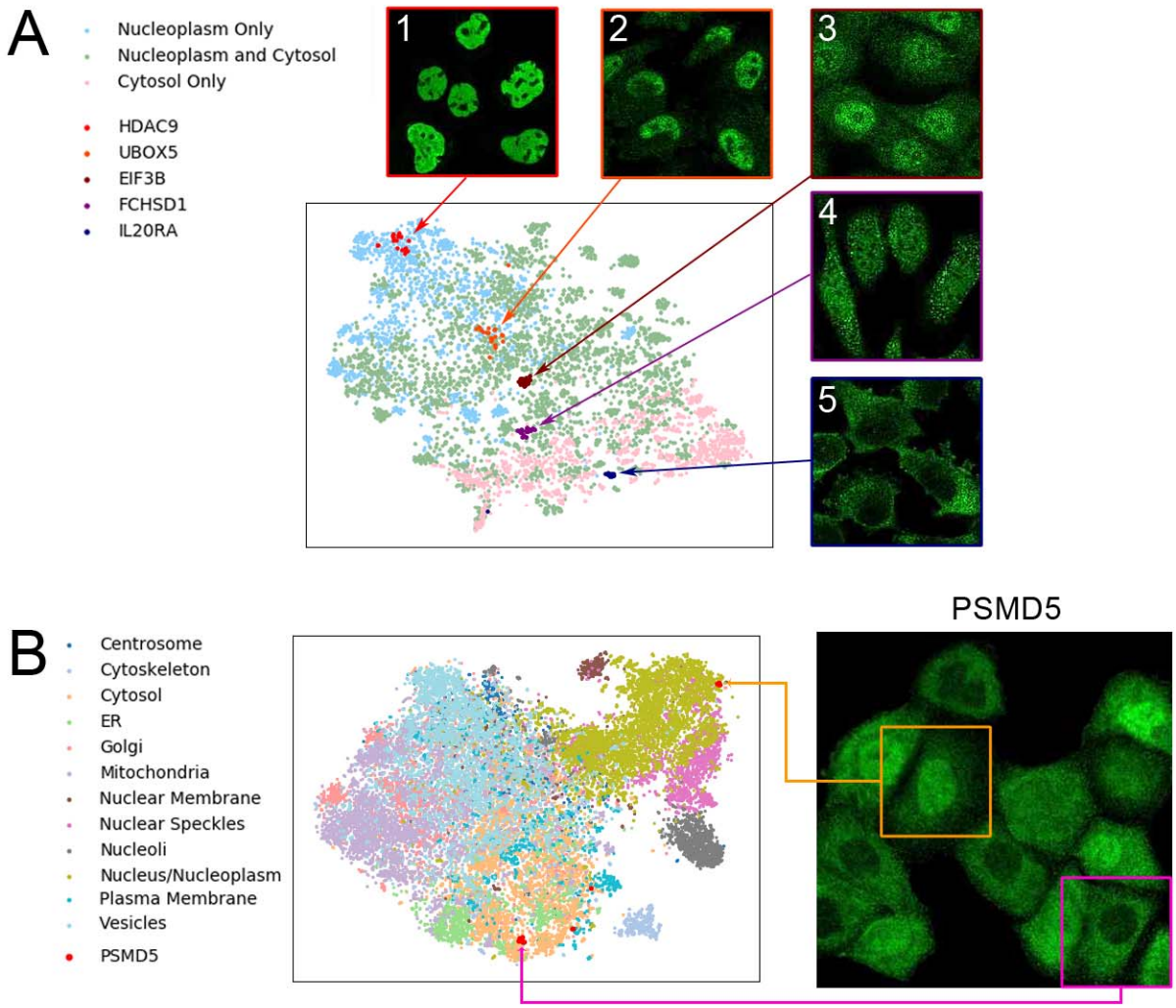


Figure 5. t-SNE scatterplots for human single cell features produced by paired cell inpainting. In A, we show single cells stained with one of 90 randomly selected proteins localizing to the nucleus and cytosol, nucleoplasm only, or cytosol only. We show cropped micrographs of the protein channel of five proteins in green in panels A-1 to A-5, as well as arrows pointing to the location of their single cells in the t-SNE scatterplot. In B, we show human A-431 single cells. In the right panel of B, we show a full image of the protein channel for PSMD5 in A-431 cells in green. We highlight the location of two single cell crops of a cell with nuclear-localized protein and one with cytosol-localized protein, in orange and purple respectively, and draw arrows pointing to their respective locations in the t-SNE scatterplot. Scatterplot points have been colored either by their annotated protein localization class or by the protein that the single cell is stained for, according to the legends. Micrographs have been rescaled by stretching the contrast to the maximum image range.

Finally, while a global analysis of proteins demonstrates that our features capture fine-grained biological insights, it does not fully exploit the resolution of our features, which are extracted for each single cell. One application of single cell features is the identification of spatially variable proteins. We reasoned that if our single cell features are sensitive to differences in protein localization, then single cells in spatially variable proteins should be distributed to different clusters. To test this, we produced a t-SNE scatterplot to visualize single cells in our A-431 evaluation set, in addition to single cells stained with an antibody for PSMD5, a protein that exhibits single-cell spatial variability between the cytosol and nucleus (Figure 5B).

In general, we observe clearly separated clusters for most classes, especially compared to CellProfiler and VGG16 features (Supplementary Figure 1). As we hypothesized, we observe that PSMD5-stained single cells that exhibit a nuclear localization cluster with other nuclear-localized single cells, while cells that exhibit a cytosol localization cluster with other cytosol-localized cells.

Discussion

We present a simple, but effective self-supervised learning task for learning feature representations of single cells in fluorescent microscopy images. Features learned through paired cell inpainting easily achieve state-of-the-art performance in unsupervised benchmarks for both yeast and human cells. We believe that the effectiveness of our learned features over classic low-level features and transfer learning can be attributed to several factors.

First, our learning task controls for nuisance variation. Because our task requires the network to predict the phenotype in a different cell, the network must ignore cell-specific variation to learn more general features about the phenotype. As we also show the network the target cell structural markers during training, the network is directly given experimental effects like local illumination, meaning that it does not have to learn features for these in the source cell encoder. Similarly, Johnson *et al*. (2017) (36) use a conditional autoencoder to disentangle variation due to the shape of the nucleus or cell from all remaining variation in the image. Their model controls for morphological variation *explicitly*: because the learning task is to reconstruct the same input image, it is necessary to split the latent image representation into separate components. In contrast, our learning task controls for nuisance variation *implicitly*. We also note that while we focus on learning unsupervised features, Johnson *et al*. focus on obtaining good generative models, so some aspects of their model are not directly applicable to our problem: particularly, their loss function requires knowing the structural class of the protein, limiting the use of unlabelled data, a key motivation of our work.

Second, self-supervised methods learn features directly from data. Here, the network learns a set of features for reconstructing spatial patterns in microscopy image datasets. Our training inputs follow the natural structure of many imaging experiments, and can be used to automatically learn features specific to each of these datasets without manual labeling. Our method is not the only one to exploit the structure of microscopy data to learn features: Caicedo *et al*. (37) propose training a CNN to classify cells by their experimental treatment. Compared to their method, ours has technical advantages. First, classifying experimental treatments may not scale as the number of experiments grows: their study uses datasets that comprise of at most, a few hundred experimental labels, but the classification problem becomes more difficult as the number of experiments grows. In our experiments, we trained cells belonging to 34,264 experiments with no complications. Second, our method does not force the CNN to discriminate between experiments that yield identical phenotypes. For example, protein complexes localize identically, so to perform well on discriminating these labels, a CNN would have to overfit to experimental variation. Our task is predictive, rather discriminative, controlling for this issue.

The features learned through paired cell inpainting may be more sensitive to biological properties in images. We performed an unbiased analysis of protein localization spanning the entire Human Protein Atlas. Multiply-localized proteins, which comprise about half of the proteins in the Human Protein Atlas (30), are challenging for classic feature sets (28), but using our features we discovered a large cluster of proteins localized to the nucleus and cytosol (cluster Q). In addition, despite pooling all cell lines and antibodies together for proteins, we capture fine-grained enrichments in our cluster analysis, that are not usually considered subcellular localizations, suggesting that images contain more functional information than discrete classes of localization.

Moreover, we showed that we could identify rare phenotypes, and discover distinct subclasses of nuclear speckles in our cluster analysis. These results emphasize the importance of unbiased analysis, in contrast to supervised approaches. Rare phenotypes may be excluded from classifiers due to lack of training data, or lack of knowledge of which images to mine training data from. We discovered three proteins with a nucleolar rim localization, based upon the one example given by the Human Protein Atlas. Similarly, pre-defined classes may hide functionally-important variability within classes, and may be biased by human pre-conceptions of the data. We showed that nuclear speck proteins can be clustered into at least two visually-distinct subclasses with different functional roles. Overall, these results demonstrate the power of unsupervised analysis in discovering new and unexpected biology, a fundamental advantage over supervised methods, no matter how accurate they are at annotating pre-defined classes.

While we focused on paired cell inpainting as a pretext task for learning features, our models also have attractive properties as generative models in synthesizing fluorescent microscopy images, similar to other generative models of single cells (35,36). When used end-to-end, the CNN can transfer protein localization from any source cell to the structure of any other cell in our dataset. This property has a variety of potential applications. For example, we can infer the localization of proteins in cell lines that have not yet been tagged for that protein. Similarly, we can infer the localization of many different proteins in the same cell, allowing us to infer co-localization in a way that would not be possible experimentally due to physical limitations on the number of fluorescent channels possible. We show an example of the latter in Figure 6: by transferring proteins from four cells stained with differently-localized antibodies onto the same source cell, we generate a prediction of a cell with the cytosol, nucleus, nucleoli, and nuclear membrane tagged simultaneously, in addition to the original antibody.

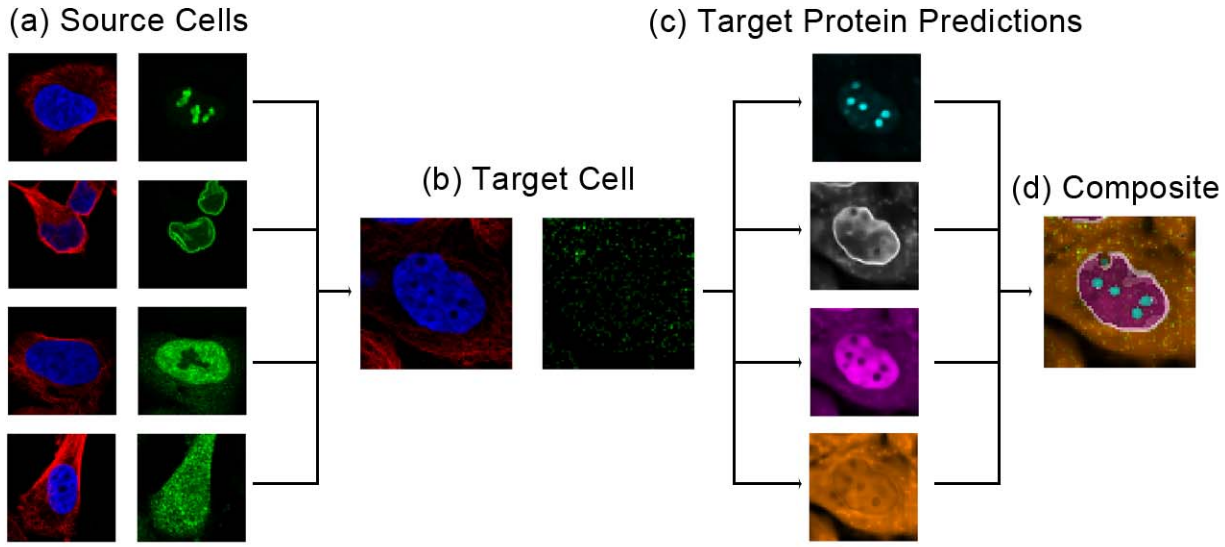


Figure 6. An example of how to use paired cell inpainting as a generative model. Here, we take four source human cells with different localizations (a), belonging in the nucleoli, nuclear membrane, nucleoplasm, and cytosol, respectively. For each of our source cells, we run our human cell model end-to-end with the same target cell marker (b) as input, generating four separate predictions (c). Finally, we merge these into a composite image showing all four predicted localizations simultaneously, in additional to the original target antibody (d). For real data, we color the nuclear channels blue, the microtubule channels red, and the protein channels green. For predicted proteins, we color nucleoli cyan, nuclear membrane white, nucleoplasm purple, and cytosol green. Channels in composite image have been rescaled and thresholded for visualization purposes.

Future work in improving some of the technical limitations of our work may further improve the quality of the learned features. First, we use an older AlexNet architecture, as we focused on the learning task and wanted to use a simple architecture with fewer parameters to reduce training time. Future work in optimizing the architecture will likely further improve the results. Second, as we discussed, some proteins localize in a pattern that cannot be predicted deterministically from the structural markers of the cell alone. The inability of the network to accurately reconstruct these proteins may limit its feature learning; incorporating an adversarial loss, or adding additional information to the inputs, may improve this. Third, for our experiments, we iterated over all cells in our training dataset and sampled pairs randomly. A more sophisticated pair sampling scheme may improve the quality of the features: for example, focusing on harder pairs where the morphology of the cells differs more drastically. Finally, there are some protein localization patterns correlated with the cell cycle, such as the mitotic spindle. Due to their low penetrance in an image (for example, only a fraction of unsynchronized cells will be undergoing mitosis), we do not expect our model to learn these patterns very well. Even if these cells are paired correctly, there may only be one or two cells with the specific cell cycle stage required to exhibit the pattern in the entire image. Better datasets with more fields of view, or work in finding an unsupervised method for balancing the patterns in the data, may improve this issue.

Code Availability

Code and pre-trained weights for the models used in this work will be made available at https://github.com/alexxijielu/paired_cell_inpainting.

Acknowledgements

We thank Shadi Zabad, Taraneh Zarin, and Purnima Kompella for valuable comments on the manuscript. This work was funded by the National Science and Engineering Research Council (Pre-Doctoral Award), Canada Research Chairs (Tier II Chair), and the Canadian Foundation for Innovation.

Contributions

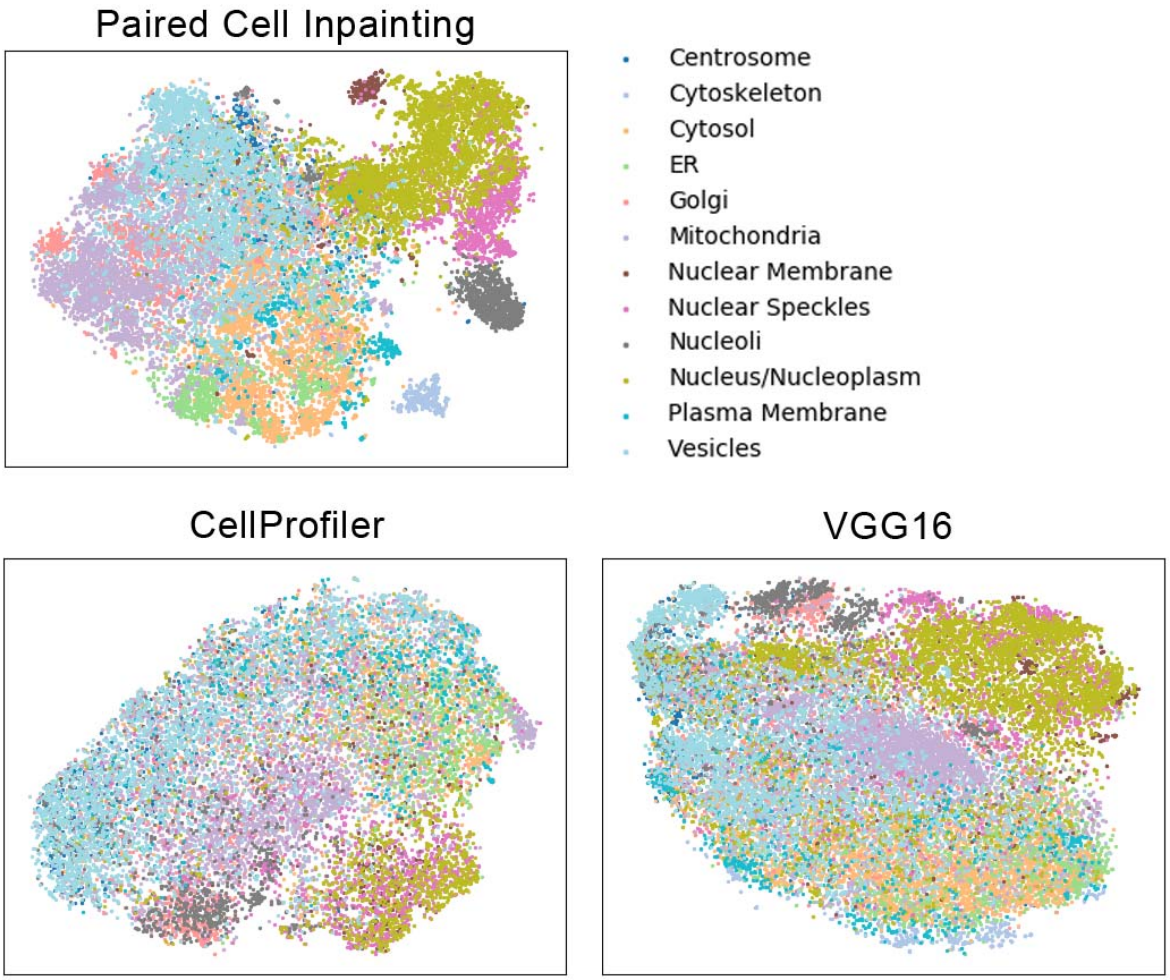
A.X.L and A.M.M conceptualized the idea and designed the experiments. A.X.L collected the data, implemented the software, and conducted the experiments and analyses. A.X.L wrote the manuscript with suggestions and edits from the other authors. O.Z.K and S.M. provided training and supervision in implementing the computational methods. A.M.M supervised the project.

References

1. Caicedo JC, Cooper S, Heigwer F, Warchal S, Qiu P, Molnar C, et al. Data-analysis strategies for image-based cell profiling. Nat Methods. Nature Publishing Group; 2017 Aug 31;14(9):849–63.
2. Chong YT, Koh JLY, Friesen H, Kaluarachchi Duffy S, Cox MJ, Moses A, et al. Yeast Proteome Dynamics from Single Cell Imaging and Automated Analysis. Cell. 2015 Jun;161(6):1413–24.
3. Lu AX, Chong YT, Hsu IS, Strome B, Handfield L-F, Kraus O, et al. Integrating images from multiple microscopy screens reveals diverse patterns of change in the subcellular localization of proteins. Elife. eLife Sciences Publications Limited; 2018 Apr 5;7:e31872.
4. Murphy RF, Porreca G. Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images *. J VLSI Signal Process. 2003;35:311–21.
5. Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. 2006 Jan;7(10):R100.
6. Handfield L-F, Chong YT, Simmons J, Andrews BJ, Moses AM. Unsupervised clustering of

subcellular protein expression patterns in high-throughput microscopy images reveals protein complexes and functional relationships between proteins. PLoS Comput Biol. 2013 Jan;9(6):e1003085.
7. Ljosa V, Caie PD, Ter Horst R, Sokolnicki KL, Jenkins EL, Daya S, et al. Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. J Biomol Screen. 2013 Dec;18(10):1321–9.
8. Kraus OZ, Grys BT, Ba J, Chong Y, Frey BJ, Boone C, et al. Automated analysis of high content microscopy data with deep learning. Mol Syst Biol. 2017 Apr 18;13(4).
9. Eulenberg P, Koehler N, Blasi T, Filby A, Carpenter AE, Rees P, et al. Deep Learning for Imaging Flow Cytometry: Cell Cycle Analysis of Jurkat Cells. bioRxiv. Cold Spring Harbor Laboratory; 2016 Oct 17;081364.
10. Dürr O, Sick B. Single-Cell Phenotype Classification Using Deep Convolutional Neural Networks. J Biomol Screen. 2016 Oct;21(9):998–1003.
11. Godinez WJ, Hossain I, Lazic SE, Davies JW, Zhang X. A multi-scale convolutional neural network for phenotyping high-content cellular images. Bioinformatics. 2017 Jul 1;33(13):2010–9.
12. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. Nature Publishing Group; 2015 May 28;521(7553):436–44.
13. Ando DM, McLean C, Berndl M. Improving Phenotypic Measurements in High-Content Imaging Screens. bioRxiv. Cold Spring Harbor Laboratory; 2017 Jul 10;161422.
14. Pawlowski N, Caicedo JC, Singh S, Carpenter AE, Storkey A. Automating Morphological Profiling with Generic Deep Convolutional Networks. bioRxiv. Cold Spring Harbor Laboratory; 2016 Nov 2;085118.
15. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA. Context Encoders: Feature Learning by Inpainting. 2016 Apr 25;
16. Jenni S, Favaro P. Self-Supervised Feature Learning by Learning to Spot Artifacts. 2018 Jun 13;
17. Gidaris S, Singh P, Komodakis N. Unsupervised Representation Learning by Predicting Image Rotations. 2018 Mar 20;
18. Zhang R, Isola P, Efros AA. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. 2016 Nov 29;
19. Kim D, Cho D, Yoo D, Kweon IS. Learning Image Representations by Completing Damaged Jigsaw Puzzles. 2018 Feb 6;
20. Mundhenk TN, Ho D, Chen BY. Improvements to context based self-supervised learning. 2017 Nov 16;
21. Waters JC. Accuracy and precision in quantitative fluorescence microscopy. J Cell Biol. 2009 Jun 29;185(7):1135–48.
22. Smith S, Grima R. Single-cell variability in multicellular organisms. Nat Commun. Nature Publishing Group; 2018 Dec 24;9(1):345.
23. Handfield L-F, Strome B, Chong YT, Moses AM. Local statistics allow quantification of cell-to-cell variability from high-throughput microscope images. Bioinformatics. Oxford University Press; 2015 Mar 15;31(6):940–7.
24. Koh JLY, Chong YT, Friesen H, Moses A, Boone C, Andrews BJ, et al. CYCLoPs: A Comprehensive Database Constructed from Automated Analysis of Protein Abundance and Subcellular Localization Patterns in Saccharomyces cerevisiae. G3 (Bethesda). 2015 Jun;5(6):1223–32.
25. Thul PJ, Åkesson L, Wiking M, Mahdessian D, Geladaki A, Ait Blal H, et al. A subcellular map of the human proteome. Science (80-). 2017;
26. van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. scikit-image: image processing in Python. PeerJ. PeerJ, Inc; 2014;2:e453.
27. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014 Sep 4;
28. Li J, Newberg JY, Uhlén M, Lundberg E, Murphy RF. Automated Analysis and Reannotation of

Subcellular Locations in Confocal Images from the Human Protein Atlas. Ranganathan S, editor. PLoS One. Public Library of Science; 2012 Nov 30;7(11):e50514.
29. Eden E, Navon R, Steinfeld I, Lipson D, Yakhini Z. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. BMC Bioinformatics. BioMed Central; 2009 Feb 3;10(1):48.
30. Thul PJ, Åkesson L, Wiking M, Mahdessian D, Geladaki A, Ait Blal H, et al. A subcellular map of the human proteome. Science (80-). 2017 May 26;356(6340):eaal3321.
31. Biggiogera M, Bürki K, Kaufmann SH, Shaper JH, Gas N, Amalric F, et al. Nucleolar distribution of proteins B23 and nucleolin in mouse preimplantation embryos as visualized by immunoelectron microscopy. Development. 1990 Dec;110(4):1263–70.
32. Lamond AI, Spector DL. Nuclear speckles: a model for nuclear organelles. Nat Rev Mol Cell Biol. 2003 Aug 1;4(8):605–12.
33. Pärnamaa T, Parts L. Accurate Classification of Protein Subcellular Localization from High-Throughput Microscopy Images Using Deep Learning. G3 (Bethesda). Genetics Society of America; 2017;7(5):1385–92.
34. Murphy RF. Building cell models and simulations from microscope images. Methods. 2016 Mar 1;96:33–9.
35. Osokin A, Chessel A, Salas REC, Vaggi F. GANs for Biological Image Synthesis. 2017 Aug 15;
36. Johnson GR, Donovan-Maiye RM, Maleckar MM. Generative Modeling with Conditional Autoencoders: Building an Integrated Cell. 2017 Apr 28;
37. Caicedo JC, McQuin C, Goodman A, Singh S, Carpenter AE. Weakly supervised learning of single-cell feature embeddings. bioRxiv. Cold Spring Harbor Laboratory; 2018 Apr 2;293431.



Supplementary Figure 1. A comparison of tSNE scatterplots for human A-431 single cell features produced by paired cell inpainting, CellProfiler, and VGG16. We color single cells by their annotated localization class in the Human Protein Atlas (as shown by the legend in the top right).