nature methods

ARTICLES

https://doi.org/10.1038/s41592-022-01541-z

Check for updates

OPEN

Self-supervised deep learning encodes high-resolution features of protein subcellular localization

Hirofumi Kobayashi ✉, Keith C. Cheveralls, Manuel D. Leonetti ✉ and Loic A. Royer ✉

Explaining the diversity and complexity of protein localization is essential to fully understand cellular architecture. Here we present cytoself, a deep-learning approach for fully self-supervised protein localization profiling and clustering. Cytoself leverages a self-supervised training scheme that does not require preexisting knowledge, categories or annotations. Training cytoself on images of 1,311 endogenously labeled proteins from the OpenCell database reveals a highly resolved protein localization atlas that recapitulates major scales of cellular organization, from coarse classes, such as nuclear and cytoplasmic, to the subtle localization signatures of individual protein complexes. We quantitatively validate cytoself's ability to cluster proteins into organelles and protein complexes, showing that cytoself outperforms previous self-supervised approaches. Moreover, to better understand the inner workings of our model, we dissect the emergent features from which our clustering is derived, interpret them in the context of the fluorescence images, and analyze the performance contributions of each component of our approach.

Systematic and large-scale microscopy-based cell assays are becoming an increasingly important tool for biological discovery[1,2], playing a key role in drug screening[3,4], drug profiling[5,6] and for mapping the subcellular localization of the proteome[7,8]. In particular, large-scale datasets based on immuno-fluorescence or endogenous fluorescent tagging comprehensively capture localization patterns across the human[9,10] and yeast proteome[11]. Together with recent advances in computer vision and deep learning[12], such datasets are poised to help systematically map the cell's spatial architecture. This situation is reminiscent of the early days of genomics, when the advent of high-throughput and -fidelity sequencing technologies was accompanied by the development of new algorithms to analyze, compare and categorize these sequences, and the genes therein. However, images pose unique obstacles to analysis. While sequences can be compared against a frame of reference (that is, genomes), there are no such references for microscopy images. Indeed, cells exhibit a wide variety of shapes and appearances that reflect a plurality of states. This rich diversity is much harder to model and analyze than, for example, sequence variability. Moreover, much of this diversity is stochastic, posing the additional challenge of separating information of biological relevance from irrelevant variance. The fundamental computational challenge posed by image-based screens is therefore to extract well-referenced vectorial representations that faithfully capture only the relevant biological information and allow for quantitative comparison, categorization and biological interpretation of protein localization patterns.

Previous approaches to classify and compare images have relied on engineered features that quantify different aspects of image content, such as cell size, shape and texture[13–16]. While these features are, by design, relevant and interpretable, the underlying assumption is that all the relevant features needed to analyze an image can be identified and appropriately quantified. This assumption has been challenged by deep learning's recent successes[17]. On a wide range of computer vision tasks such as image classification, hand-designed features cannot compete against learned features that are automatically discovered from the data themselves[18,19]. Assuming features are available, the typical approach consists of boot-strapping the annotation process by either (1) unsupervised clustering techniques[20,21], or (2) manual curation and supervised learning[22,23]. In the case of supervised approaches, human annotators examine images and assign annotations, and once sufficient data are garnered, a machine learning model is trained in a supervised manner and later applied to unannotated data[17,18,23,24]. Another approach consists of reusing models trained on natural images to learn generic features on which supervised training can be bootstrapped[5,25,26]. While successful, these approaches suffer from potential biases, as manual annotation imposes our own preconceptions. Overall, the ideal algorithm should not rely on human knowledge or judgments, but instead automatically synthesize features and analyze images without a priori assumptions, that is, solely on the basis of the images themselves.

Recent advances in computer vision and machine learning have shown that forgoing manual labeling is possible and nears the performance of supervised approaches[27,28]. Instead of annotating datasets, which is inherently nonscalable and labor-intensive, self-supervised models can be trained from large uncurated datasets[11,29–32]. Self-supervised models are trained by formulating an auxiliary pretext task, typically one that withholds parts of the data and instructs the model to predict them[33]. This works because the task-relevant information within a piece of data is often distributed over multiple observed dimensions[30]. For example, given the picture of a car, we can recognize the presence of a vehicle even if many pixels are hidden, perhaps even when half of the image is occluded. Now, consider a large dataset of pictures of real-world objects (for example, ImageNet[34]). Training a model to predict hidden parts from these images forces it to identify their important features[32]. Once trained, the vectorial representations that emerge from pretext tasks capture the important features of the images, and can be used for comparison and categorization.

Here, we present the development, validation and use of cytoself, a deep learning-based approach for fully self-supervised

Chan Zuckerberg Biohub, San Francisco, CA, USA. ✉e-mail: hirofumi.kobayashi@czbiohub.org; manuel.leonetti@czbiohub.org; loic.royer@czbiohub.org

protein localization profiling and clustering. The key innovation is a pretext task that ensures that the localization features that emerge from different images of the same protein are helpful to distinguish the microscopy images of that protein from the images of other proteins in the dataset. We demonstrate the ability of cytoself to reduce images to feature profiles characteristic of protein localization, validate their use to predict protein assignment to organelles and protein complexes, and compare the performance of cytoself with previous image featurization approaches.

Results

A robust and comprehensive image dataset. A prerequisite to our deep-learning approach is a collection of high-quality images of fluorescently tagged proteins obtained under uniform conditions. Our OpenCell[10] dataset of live-cell confocal images of 1,311 endogenously tagged proteins (http://opencell.czbiohub.org) meets this purpose. We reasoned that providing a fiducial channel could provide a useful reference frame for our model to capture protein localization. Hence, in addition to imaging the endogenous tag (split mNeonGreen2), we also imaged a nuclear fiducial marker (Hoechst 33342) and converted it into a distance map (Methods). On average, we imaged the localization of a given protein in 18.59 fields of view (FOV). Approximately 45 cropped images from each FOV containing 1–3 cells were then extracted for a total of 800 cropped images per protein. This scale, as well as the uniform conditions under which the images were collected, were important because our model must learn to ignore irrelevant image variance and instead focus on protein localization. Finally, in our approach all images that represent the same protein were labeled by the same unique identifier (we used the corresponding synthetic cell line identifier, but the identifier may be arbitrary). This identifier does not carry any explicit localization information, nor is it linked to any metadata or annotations, but rather is used to link together all the different images of the same protein.

A deep-learning model to generate image representations. Our deep-learning model is based on the vector quantized variational autoencoder architecture (VQ-VAE[35,36]). In a classical VQ-VAE, images are encoded into a quantized latent representation, a vector, and then decoded to reconstruct the input image (Fig. 1 and Supplementary File 1). The encoder and decoder are trained so as to minimize distortion between input and output images. The representation produced by the encoder is assembled by arraying a finite number of symbols (indices) that stand for vectors in a codebook (Fig. 1b and Supplementary Fig. 1). The codebook vectors themselves evolve during training so as to be most effective for the encoding–decoding task[35]. The latest incarnation of this architecture (VQ-VAE-2, ref. [37]) introduces a hierarchy of representations that operate at multiple spatial scales (termed VQ1 and VQ2 in the original VQ-VAE-2 study). We chose this architecture as a starting point because of the large body of evidence that suggests that quantized architectures currently learn the best image representations[35,36]. As shown in Fig. 1b, we developed a variant that uses a split vector quantization scheme to improve quantization at large spatial scales (Methods and Supplementary Fig. 1). This new approach to vector quantization achieves better perplexity as shown in Fig. 1c, which means better codebook use.

Protein localization encoding via self-supervision. Our model consists of two pretext tasks applied to each individual cropped image: first, it is tasked to encode and then decode the image as in the original VQ-VAE model. Second, it is tasked to predict the protein identifier associated with the image solely on the basis of the encoded representation. In other words, that second task aims to predict, for each single cropped image, which one of the 1,311 proteins in our library the image corresponds to. The first task forces our model to distill lower-dimensional representations of the images, while the second task forces these representations to be strong predictors of protein identity. This second task assumes that protein localization is the primary image information that is correlated to protein identity. Therefore, predicting the identifier associated with each image is key to encouraging our model to learn localization-specific representations. It is acceptable, and in some cases perfectly reasonable, for these tasks to fail. For example, when two proteins have identical localization, it is impossible to resolve the identity of the tagged proteins from images alone. Moreover, the autoencoder might be unable to perfectly reconstruct an image from the intermediate representation, when constrained to make that representation maximally predictive of protein identity. It follows that the real output of our model is not the reconstructed image, nor the predicted identity of the tagged protein, but instead the distilled image representations, which we refer to as 'localization encodings', that are obtained as a necessary byproduct of satisfying both pretext tasks. Specifically, our model encodes two representations for each image that correspond to two different spatial scales, the local and global representations, that correspond to VQ1 and VQ2, respectively. The global representation captures large-scale image structure scaled-down to a 4×4 pixel image with 576 features (values) per pixel. The local representation captures finer spatially resolved details (25×25 pixel image with 64 features per pixel). We use the global representations to perform localization clustering, and the local representations to provide a finer and spatially resolved decomposition of protein localization.

Mapping the protein localization landscape with cytoself. Obtaining image representations that are highly correlated with protein localization and invariant to other sources of heterogeneity (that is, cell state, density and shape) is only the first step for biological interpretation. Indeed, while these representations are lower dimensional than the images themselves, they still have too many dimensions for direct inspection and visualization. Therefore, we performed dimensionality reduction using the uniform manifold approximation and projection (UMAP) algorithm on the set of global localization encodings (that is, global representation in the Fig. 1) obtained from all images (Methods). In the resulting UMAP (Fig. 2) each point represents a single (cropped) image in our test dataset (that is, 10% of entire dataset, Methods), which collectively form a highly detailed map representing the full diversity of protein subcellular localizations. This protein localization atlas reveals an organization of clusters and subclusters reflective of eukaryotic subcellular architecture. We can evaluate and explore this map by labeling each protein according to its known subcellular localization obtained from independent manual annotations of our image dataset (Supplementary File 2). The most pronounced delineation corresponds to nuclear (top right) versus nonnuclear (bottom left) localizations (encircled and expanded in Fig. 2, top right and bottom left, respectively). Within the nuclear cluster, subclusters are resolved that correspond to nucleoplasm, chromatin, nuclear membrane and the nucleolus. Within each region, tight clusters that correspond to specific cellular functions can be resolved (dashed outlines). For example, subunits involved in splicing (SF3 spliceosome), transcription (core RNA polymerase) or nuclear import (nuclear pore) cluster tightly together (outlined in Fig. 2, dashed outlines). Similarly, subdomains emerge within the nonnuclear cluster, the largest corresponding to cytoplasmic and vesicular localizations. Within these domains are several well delineated clusters corresponding to mitochondria, endoplasmic reticulum (ER) exit sites (COPII), ribosomes and clathrin coated vesicles (Fig. 2). The large set of unlabeled points in Fig. 2 (gray dots) correspond mainly to proteins that exhibit mixed localization patterns. Prominent among these is a band of proteins interspersed between the nuclear and nonnuclear regions (expanded in Fig. 3a). Representative

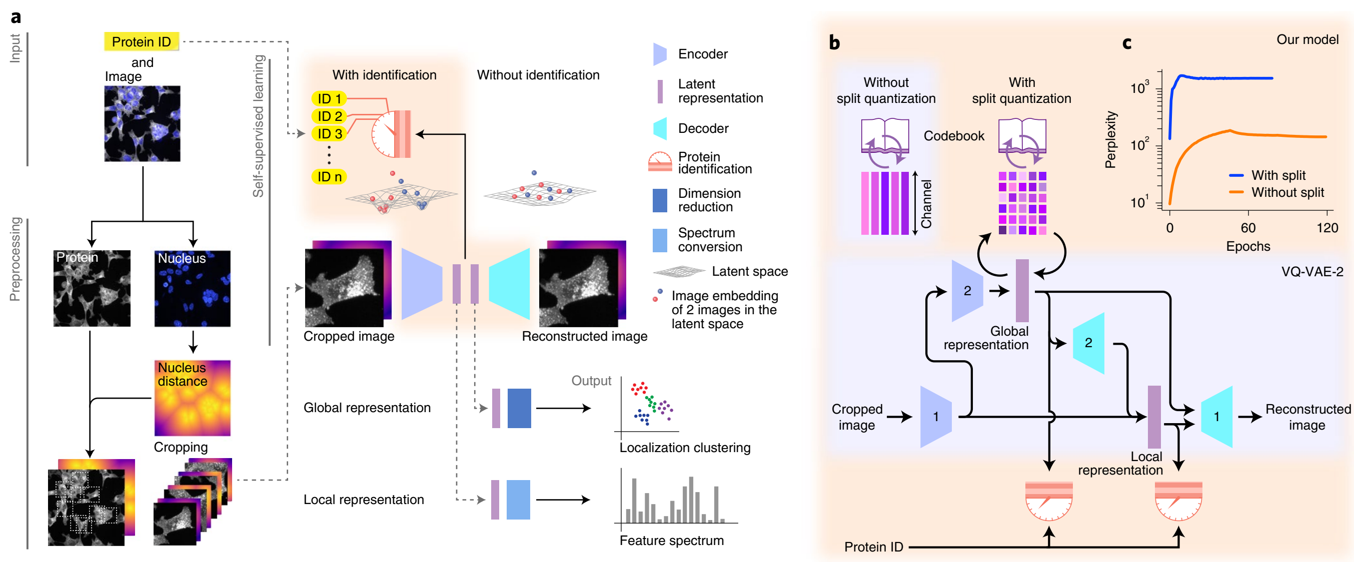


Fig. 1 | Self-supervised deep learning of protein subcellular localization with cytoself. a, Workflow of the learning process. Only images and the proteins identifiers are required as input. We trained our model with a second fiducial channel for the cell nuclei, but its presence is optional as its performance contribution is negligible (Fig. 4). The protein identification pretext task ensures that images corresponding to the same or similar proteins have similar representations. **b**, Architecture of our VQ-VAE-2 (ref. [37]) -based deep-learning model featuring our two innovations: split-quantization and protein identification pretext task. Numbers in the encoders and decoders indicate encoder1, encoder2, decoder1 or decoder2 (Supplementary File 1). Global representation and local representation use different codebooks. **c**, The level of use of the codebook (that is, perplexity) increases and then saturates during training and is enhanced by applying split quantization.

proteins chosen along that path show a continuous gradation from mostly cytoplasmic to mostly nuclear localization.

Quantifying cytoself's clustering performance. To validate our results, clustering scores were computed (Methods, Fig. 4 and Table 1) using two ground-truth annotation datasets to capture known protein localization at two different scales: the first is a manually curated list of proteins with unique organelle-level localizations (Supplementary File 2), whereas the second is a list of proteins participating in stable protein complexes derived from the CORUM database[38] (Supplementary File 3). While the first ground-truth dataset helps us assess how well our encodings cluster together proteins belonging to the same organelles, the second helps us assess whether proteins interacting within the same complex—and thus functionally related—are in proximity. We compared cytoself to other previously developed unsupervised (CellProfiler[14]) or self-supervised (Cell inpainting[11]) approaches for image featurization. We applied these methods to the OpenCell image dataset and then compared the results to that obtained by cytoself. UMAPs were calculated for each model (Methods) and compared with our set of ground-truth organelles and protein complexes. As can be seen in Extended Data Figs. 1 and 2 and Supplementary Fig. 2, the resolution obtained by cytoself exceeded that of both previous approaches. This was also apparent in our calculations of clustering scores (Fig. 4 and Table 1).

Identifying cytoself's essential components. To evaluate the impact of different aspects of our model on its clustering performance, we conducted an ablation study. We retrained our model and recomputed protein localization UMAPs after individually removing each component or input of our model (Extended Data Figs. 3 and 4), including: (1) the nuclear fiducial channel, (2) the distance transform applied to nuclear fiducial channel, (3) the split vector quantization and (4) the identification pretext task. We also quantitatively evaluated the effects of their ablation by computing clustering scores for different variants (Fig. 4 and Table 1). The UMAP results and scores from both sets of ground-truth labels make it clear that the single most important component of cytoself, in terms of clustering performance, is the protein identification pretext task. The remaining components—the nuclear channel, split quantization, vector quantization and so on—are important but not crucial. Forgoing the fiducial nuclear channel entirely led to the smallest decrease in clustering score, suggesting that our approach works well even in the absence of any fiducial marker—a notable advantage that widens the applicability of our approach and greatly simplifies the experimental design[39]. Overall, our data show a robust fit with ground truth. In conclusion, although all features contribute to the overall performance of our model, the identification pretext task is the key and necessary ingredient.

Revealing unannotated protein localization. The key advantage of self-supervised approaches is that they are not limited by the quality, completeness or granularity of human annotations. To demonstrate this, we asked whether cytoself could resolve subtle localization differences that are not present in image-derived manual annotations: focusing on proteins localized to intracellular vesicles. Even though several known subcategories of vesicles exist (for example, lysosomes versus endosomes), in both OpenCell and Human Protein Atlas annotations, these groups are annotated simply as 'vesicles'. This reflects the difficulty for human curators to accurately distinguish and classify localization subcategories that present similarly in the images. To test whether our self-supervised approach manages to capture these subcategories, we focused on a curated list of endosomal as well as lysosomal proteins identified by an objective criterion. Specifically, we selected proteins annotated as lysosomal (GO 000576500) or endosomal (GO 0031901) in Uniprot[40] (excluding targets annotated to reside in both compartments), and for which localization in each compartment has been confirmed independently by mass spectrometry[41,42]. As shown in Extended Data Fig. 5, the representation of the lysosomal versus endosomal

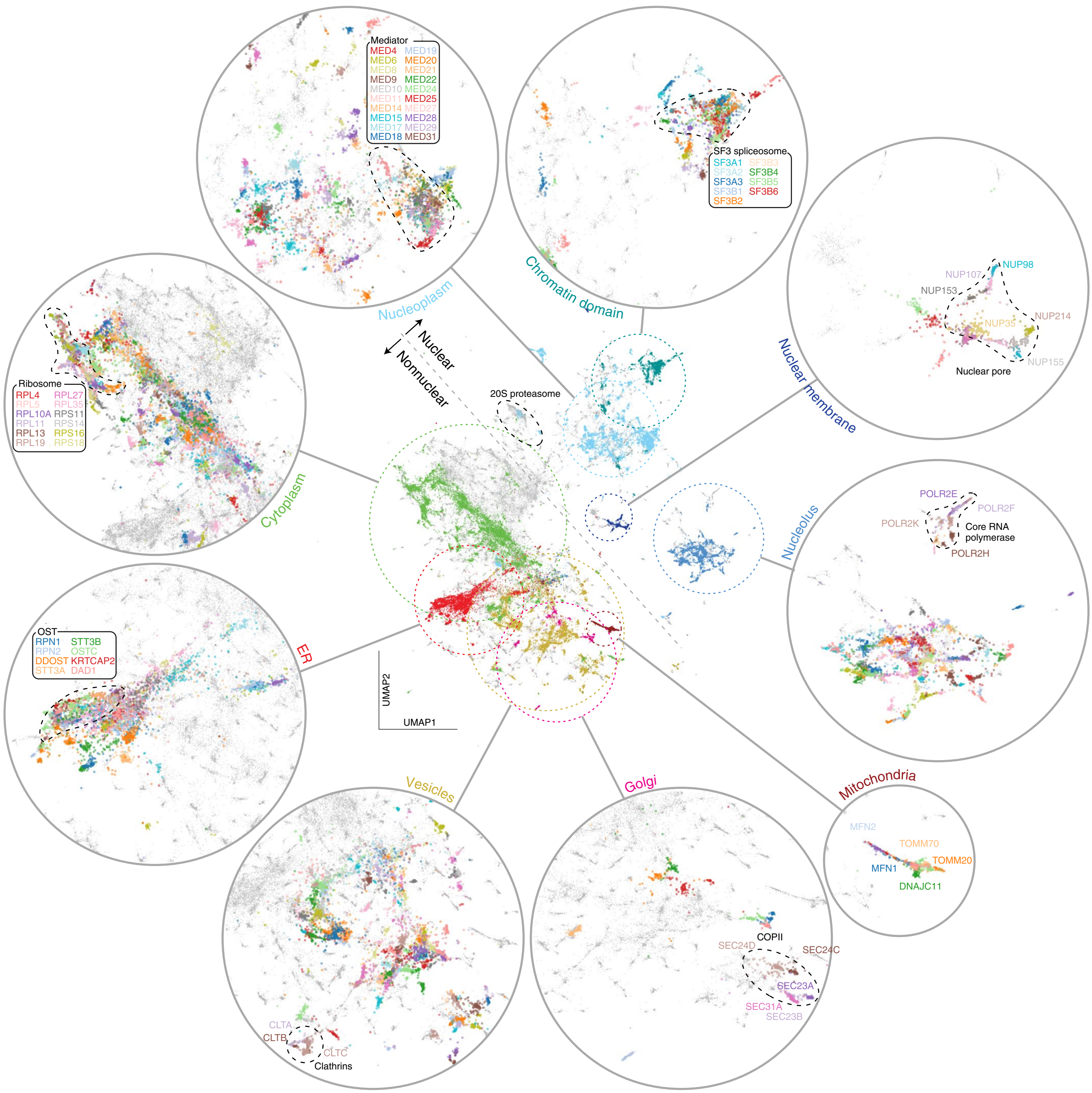


Fig. 2 | High-resolution protein localization atlas. Each point corresponds to a single image from our test dataset of 109,751 images. To reveal the underlying structure of our map, each point in the central UMAP is colored according to 11 distinct protein localization categories (mitochondria, vesicles, nucleoplasm, cytoplasm, nuclear membrane, ER, nucleolus, Golgi, chromatin domain). These categories are expanded in the surrounding circles. Tight clusters corresponding to functionally defined protein complexes can be identified within each localization category. Only proteins with a clear and exclusive localization pattern are colored, gray points correspond to proteins with other or mixed localizations. Within each localization category, the resolution of cytoself representations is further illustrated by labeling the images corresponding to individual proteins in different colors (dashed circular inserts). Note that while the colors in the central UMAP represent different cellular territories, colors in the inserts are only used to delineate individual proteins, and do not correspond to the colors used in the main UMAP. The list of annotated proteins and the subunits of each complex are indicated in Supplementary Files 2 and 5, respectively.

images derived from cytoself form two distinct, well-separated clusters ($P < 10^{-3}$, Mann–Whitney U-test). This demonstrates that self-supervised approaches are not limited by ground-truth annotations and can reveal subtle differences in protein localization not explicitly present in existing databases.

Extracting feature spectra for quantitative analysis. Cytoself can generate a highly resolved map of protein localization on the basis of distilled image representations. Can we dissect and understand the features that make up these representations and interpret their meaning? To identify and better define the features that make up

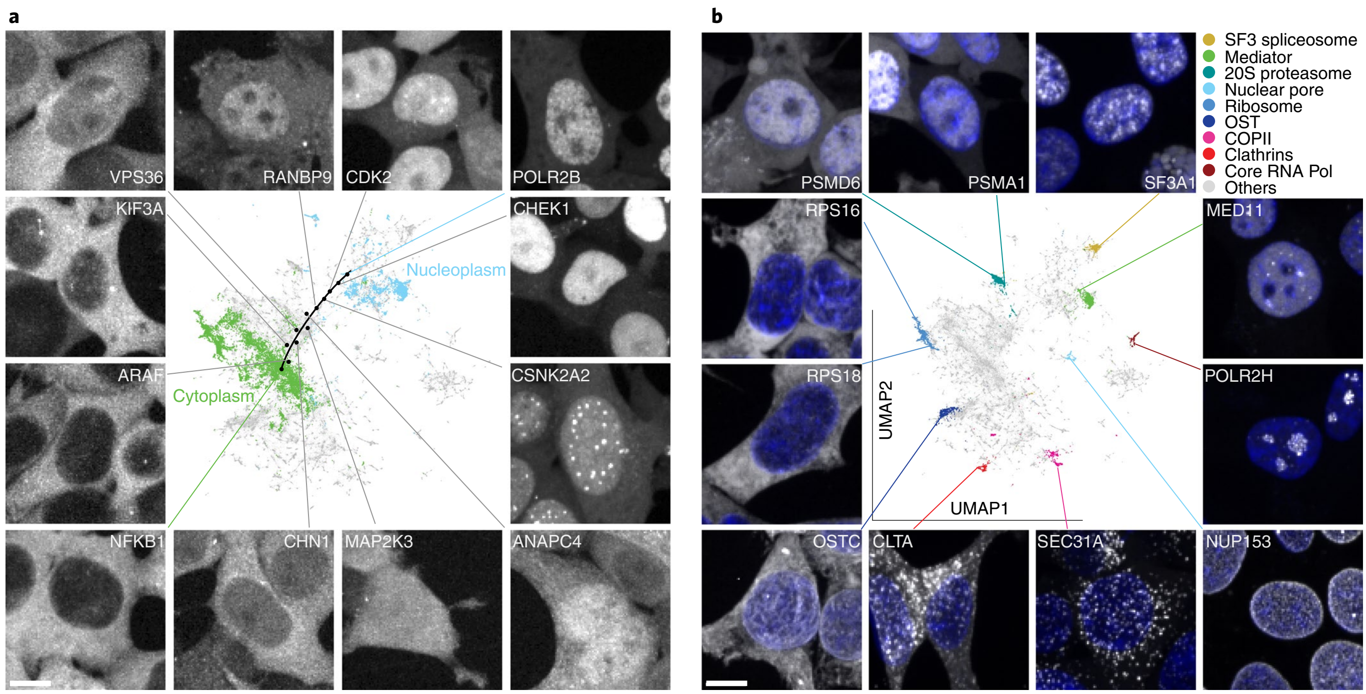


Fig. 3 | Exploring the protein localization atlas. a, Representative images of proteins localized along an exemplary path across the nuclear-cytoplasmic transition and over the 'gray' space of mixed localizations. **b**, The subunits of well-known and stable protein complexes tightly cluster together. Moreover, the complexes themselves are placed in their correct cellular contexts. Different proteins have different expression levels, hence we adjusted the brightness of each panel so as to make all localizations present in each image more visible (only minimum–maximum intensities are adjusted, no gamma adjustment used). All representative images were randomly selected. Protein localization is displayed in grayscale in both panels, the nuclei in **b** are displayed in blue. The list of the subunits of each complex are indicated in Supplementary File 5. Scale bars, 10 µm.

these representations, we created a feature spectrum of the main components contributing to each protein's localization encoding. The spectra were constructed by calculating the histogram of codebook feature indices from the local representations in Fig. 1 (see Extended Data Fig. 6 and Methods for details). To group related and possibly redundant features together, we performed hierarchical biclustering[43] (Fig. 5a), and thus obtained a meaningful linear ordering of features by which the spectra can be sorted. This analysis reveals feature clusters of which we manually selected 11 from the top levels of the feature hierarchy (Fig. 5a, bottom and Supplementary Fig. 3).

Representative images from each cluster illustrate the variety of distinctive localization patterns that are present at different levels across all proteins. For example, the features in the first clusters (i, ii, iii and iv) correspond to a wide range of diffuse cytoplasmic localizations. Cluster v features are unique to nucleolar proteins. Features making up cluster vi correspond to very small and bright punctate structures that are often characteristic of centrosomes, vesicles or cytoplasmic condensates. Clusters vii, viii and x correspond to different types of nuclear localization pattern. Cluster ix are dark features corresponding to nonfluorescent background regions. Finally, cluster xi corresponds to a large variety of more abundant, punctate structures occurring throughout the cells, primarily vesicular, but also Golgi, mitochondria, cytoskeleton and subdomains of the ER. For a quantitative evaluation, we computed the average feature spectrum for all proteins belonging to each localization category present in our reference set of manual annotations (for example, Golgi, nucleolus and so on; Fig. 5b and Supplementary File 4). This analysis confirms that certain spectral clusters are specific to certain localization categories and thus correspond to characteristic textures and patterns in the images. For example, the highly specific chromatin and mitochondrial localizations both appear to elicit very narrow responses in their feature spectra.

Predicting protein organelle localization with cytoself. We next asked whether feature spectra could be used to predict the localizations of proteins not present in our training data. For this purpose, we computed the feature spectrum of FAM241A: a protein of unknown function that was not present in the training dataset. Its spectrum is most correlated to the consensus spectrum of proteins belonging to the ER (Fig. 5b–d and Supplementary Fig. 4). Indeed, FAM241A's localization to the ER was validated experimentally by coexpression experiments showing that endogenously tagged FAM241A colocalizes with an ER marker (Fig. 5e). In a companion study[10], we further validated by mass spectrometry that FAM241A is in fact a new subunit of the oligosaccharyltransferase complex, responsible for cotranslational glycosylation at the ER membrane. Our successful prediction of the localization of FAM241A suggests that cytoself encodings can be used more generally to predict organelle-level localization categories. To demonstrate this, we focused on proteins annotated to localize to a single organelle (that is, not multi-localizing, Supplementary File 4). For each of these proteins, we recomputed the representative spectra for each of their known localization categories (that is, ER, mitochondria, Golgi and so on), but leaving out that protein, and then applied the same spectral correlation as described for FAM241A. This allows us to predict the protein's localization by identifying the organelle with which its spectrum correlates best. Extended Data Fig. 7 shows the accuracy of the predictions derived from this approach: for 88% of proteins, the spectra correlate best with the correctly annotated organelle. For 96% of proteins, the correct annotation is within the top two predictions and for 99% it is within the top three predictions. Overall, this form of cross-validation verifies the discriminating power of our

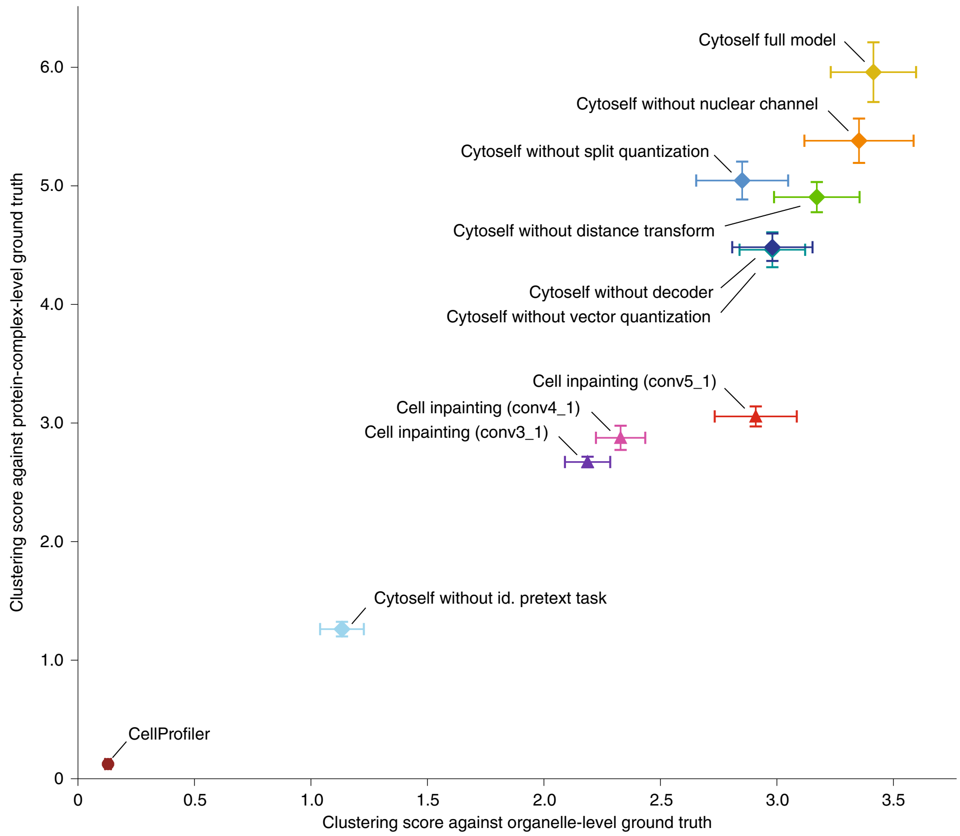


Fig. 4 | Clustering performance comparison. For each model variation, we trained five model instances, compute UMAPs for ten random seeds, compute clustering scores using organelle- and protein-complex-level ground truth and then report the mean and standard error of the mean.

Table 1 | Clustering performance comparison

Approach	Organelle level	Complex level
Cytoself full model	3.41 ± 0.18	5.96 ± 0.25
Without nuclear channel	3.35 ± 0.23	5.38 ± 0.19
Without distance transform	3.17 ± 0.18	4.90 ± 0.13
Without vector quantization	2.98 ± 0.14	4.46 ± 0.15
Without id. pretext task	1.13 ± 0.094	1.26 ± 0.062
Without split quantization	2.85 ± 0.20	5.04 ± 0.16
Without decoder	2.98 ± 0.17	4.48 ± 0.12
Lu et al. (conv3_1)	2.19 ± 0.097	2.67 ± 0.045
Lu et al. (conv4_1)	2.33 ± 0.11	2.88 ± 0.10
Lu et al. (conv5_1)	2.91 ± 0.18	3.06 ± 0.084
CellProfiler	0.129 ± 0.013	0.124 ± 0.0074

Our full model surpasses all variants considered, the previously reported cell-inpainting model[11] and CellProfiler derived representations[14]. We trained the models five times, computed ten different UMAPs, computed clustering scores using organelle- and protein-complex-level ground truth, and then report the mean and standard error of the mean (mean ± s.e.m.). For the latent representations in the inpainting model, we examined the three network layers discussed in Lu et al. to produce image representations for UMAP. Note that our approach works with a single fluorescence channel whereas the approach by Lu et al. needs at least two channels.

spectra and shows that the information encoded in each protein's spectrum can be interpreted to predict subcellular localization.

Cytoself applicability beyond OpenCell data. Can cytoself make reasonable protein localization predictions for images from datasets other than OpenCell? To answer this question, we chose data from the Allen Institute Cell collection[44], which also uses endogenous tagging and live-cell imaging, making their image data directly comparable to ours. The Allen collection uses a cell line (WTC11, induced pluripotent stem cell) whose overall morphology is very different from the cell line used for OpenCell (human embryonic kidney 293T: HEK293T). We reasoned that if cytoself manages to capture true features of protein localization, a compelling validation would be that its performance would be cell-type agnostic. Indeed, localization encodings for images from the Allen dataset generated by a cytoself model trained only on OpenCell images revealed a strong concordance between the embeddings of the same (or closely related) protein that were imaged in both cell datasets (Extended Data Fig. 8a). This shows that our model manages to predict protein localization even under conditions that were not directly included for training. To facilitate comparison, we focused on the intersection set of nine proteins found in both the OpenCell and Allen datasets (Extended Data Fig. 8b). We ran the same organelle localization prediction task and observed that in 88% (eight out of nine) of cases the correct localization is among the top three predictions (Supplementary Fig. 5).

Hypothesizing protein-complex membership from images. The resolving power of our approach is further illustrated by examining known stable protein complexes, which are found to form well delineated clusters in our localization UMAP (see examples highlighted in Fig. 2, dashed line). Fluorescent images of 11 representative subunits from these complexes illustrate these discrete localization patterns (Fig. 3b). To substantiate these observations quantitatively, we computed the correlation of feature spectra between any two pairs of proteins in our dataset. This showed a significantly higher correlation for protein pairs annotated to belong to the same complex in CORUM compared to pairs that are not ($P < 10^{-10}$, Mann–Whitney *U*-test; Supplementary Fig. 6a). To further evaluate the relationship between proximity in feature space and protein-complex membership, we examine the proportion of proteins in OpenCell that share complex membership with their most-correlated neighboring protein (Supplementary Fig. 6b). We find that 83% of highly correlated (>0.95) neighbor proteins are in the same complex, and even more weakly correlated (>0.8) proteins are localized to complexes 60% of the time. These results confirm that close proximity in feature space is highly indicative of protein-complex membership and suggests that the features derived by cytoself contain fine-grained information related to very specific functional relationships.

Discussion

We have shown that a self-supervised training scheme can produce image representations that capture the organization of protein subcellular localization (Fig. 2), solely on the basis of a large high-quality dataset of fluorescence images. Our model generates a high-resolution localization atlas capable of delineating not only organelles, but also protein complexes. Moreover, we can represent each image with a feature spectrum to better analyze the repertoire of localization patterns present in our data. Since a protein's localization is highly correlated with its cellular function, cytoself will be an invaluable tool to make preliminary functional predictions for unknown or poorly studied proteins, and for quantitatively studying the effect of cellular perturbations and cell state changes on protein subcellular localization.

Our method makes few assumptions, but imposes two pretext tasks (that is, image and protein identity). Of these, requiring the model to identify proteins based solely on their localization encodings was essential. We also included Hoechst DNA-staining as a fiducial marker, assuming that this would provide a spatial reference frame against which to interpret localization. However, this added little to the performance of our model in terms of clustering score. By comparison, the self-supervised approach by Lu et al.[11] applied a pretext task that predicts the fluorescence signal of a labeled protein in one cell from its fiducial markers and from the fluorescence signal in a second, different cell from the same FOV. This assumes that fiducial channels are available, and that protein fluorescence is always well-correlated to these fiducials. In contrast, our approach only requires a single fluorescence channel and yields better clustering performance (Fig. 4 and Table 1).

The main difference between our work and the problem addressed by the Human Protein Atlas Image Classification competition[23] is that we do not aim to predict localization patterns on the

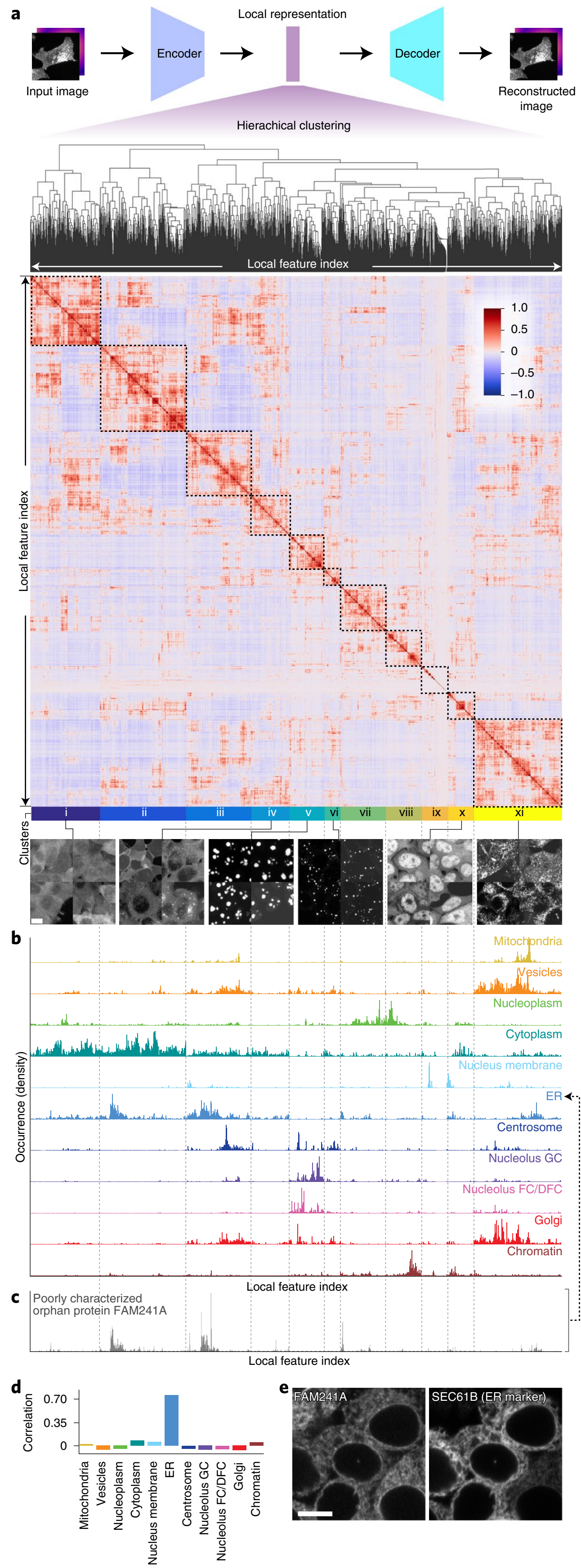


Fig. 5 | Feature spectral analysis. **a**, Features in the local representation are reordered by hierarchical clustering to form a feature spectra (Extended Data Fig. 6). The color bar indicates the strength of correlation. Negative values indicate anti-correlation. On the basis of the feature clustering, we manually identified 11 primary top-level clusters, which are illustrated with representative images (Supplementary Fig. 3). Those images have the highest occurrence of the corresponding features. **b**, Average feature spectrum for each unique localization family. Occurrence indicates how many times a quantized vector is found in the local representation of an image. All spectra, as well as the heatmap, are vertically aligned. **c**, The feature spectrum of FAM241A, a poorly characterized orphan protein. **d**, Correlation between FAM241A and other unique localization categories. The highest correlation is 0.777 with ER, next is 0.08 with cytoplasm. **e**, Experimental confirmation of the ER localization of FAM241A. The localization of FAM241A to the ER is experimentally confirmed by coexpression of a classical ER marker (mCherry fused to the SEC61B transmembrane domain, left) in FAM241A-mNeonGreen endogenously tagged cells (right). The ER marker is expressed using transient transfection. As a consequence, not all cells are transfected and levels of expression may vary. Scale bars, 10 µm.

basis of manual annotations. Instead, we aim to discover de novo the landscape of possible protein localizations. This frees us from the limitations of these annotations that include: lack of uniform coverage, uneven annotation granularity, human perceptive biases and existing preconceptions on the architecture of the cell. This also circumvents the time-intensive efforts required to manually annotate images.

While powerful, there remain a few avenues for further development of cytoself. For example, we trained our model using two-dimensional (2D) maximum-intensity z-projections and have not yet leveraged the full three-dimensional (3D) confocal images available in the OpenCell[10] dataset. The third dimension might confer an advantage for specific protein localization patterns that are characterized by specific variations along the basal-apical cell axis. Other important topics to explore are the automatic suppression of residual batch effects, improved cell segmentation via additional fiducial channels, use of label-free imaging modalities, as well as automatic rejection of anomalous or uncharacteristic cells from our training dataset. More fundamentally, notable conceptual improvements will require an improved self-supervised model that explicitly disentangles cellular heterogeneity from localization diversity[45].

More generally, our ability to generate data is outpacing the human ability to manually annotate it. Moreover, there is already ample evidence that abundance of image data has a quality all its own: Increasing the size of an image dataset often has a higher impact on performance than improving the algorithm itself[46]. We envision that self-supervision will be a powerful tool to handle the large amount of data produced by new instruments, end-to-end automation and high-throughput image-based assays.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-022-01541-z.

Received: 2 July 2021; Accepted: 26 May 2022;
Published online: 25 July 2022

References

1. Pepperkok, R. & Ellenberg, J. High-throughput fluorescence microscopy for systems biology. *Nat. Rev. Mol. Cell Biol.* **7**, 690–696 (2006).
2. Chandrasekaran, S. N., Ceulemans, H., Boyd, J. D. & Carpenter, A. E. Image-based profiling for drug discovery: due for a machine-learning upgrade?. *Nat. Rev. Drug Discov.* **20**, 145–159 (2020).
3. Boutros, M., Heigwer, F. & Laufer, C. Microscopy-based high-content screening. *Cell* **163**, 1314–1325 (2015).
4. Abraham, V. C., Taylor, D. L. & Haskins, J. R. High content screening applied to large-scale cell biology. *Trends Biotechnol.* **22**, 15–22 (2004).
5. Scheeder, C., Heigwer, F. & Boutros, M. Machine learning and image-based profiling in drug discovery. *Curr. Opin. Syst. Biol.* **10**, 43–52 (2018).
6. Loo, L.-H., Wu, L. F. & Altschuler, S. J. Image-based multivariate profiling of drug responses from single cells. *Nat. Methods* **4**, 445–453 (2007).
7. Huh, W.-K. et al. Global analysis of protein localization in budding yeast. *Nature* **425**, 686–691 (2003).
8. Cai, Y. et al. Experimental and computational framework for a dynamic protein atlas of human cell division. *Nature* **561**, 411–415 (2018).
9. Thul, P. J. et al. A subcellular map of the human proteome. *Science* **356**, aal3321 (2017).
10. Cho, N. H. et al. Opencell: endogenous tagging for the cartography of human cellular organization. *Science* **375**, eabi6983 (2022).
11. Lu, A. X., Kraus, O. Z., Cooper, S. & Moses, A. M. Learning unsupervised feature representations for single cell microscopy images with paired cell inpainting. *PLoS Computat. Biol.* **15**, e1007348 (2019).
12. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
13. Perlman, Z. E. et al. Multidimensional drug profiling by automated microscopy. *Science* **306**, 1194–1198 (2004).
14. Carpenter, A. E. et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006).
15. Yin, Z. et al. A screen for morphological complexity identifies regulators of switch-like transitions between discrete cell shapes. *Nat. Cell Biol.* **15**, 860–871 (2013).
16. Bray, M.-A. et al. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nat. Protoc.* **11**, 1757 (2016).
17. Kraus, O. Z. et al. Automated analysis of high-content microscopy data with deep learning. *Mol. Syst. Biol.* **13**, 924 (2017).
18. Eulenberg, P. et al. Reconstructing cell cycle and disease progression using deep learning. *Nat. Commun.* **8**, 463 (2017).
19. Caicedo, J. C. et al. Data-analysis strategies for image-based cell profiling. *Nat. Methods* **14**, 849–863 (2017).
20. Sailem, H., Bousgouni, V., Cooper, S. & Bakal, C. Cross-talk between rho and RAC GTPases drives deterministic exploration of cellular shape space and morphological heterogeneity. *Open Biol.* **4**, 130132 (2014).
21. Traag, V. A., Waltman, L. & Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).
22. Jones, T. R. et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proc. Natl Acad. Sci. USA* **106**, 1826–1831 (2009).
23. Ouyang, W. et al. Analysis of the human protein atlas image classification competition. *Nat. Methods* **16**, 1254–1261 (2019).
24. Blasi, T. et al. Label-free cell cycle analysis for high-throughput imaging flow cytometry. *Nat. Commun.* **7**, 10256 (2016).
25. Pawlowski, N., Caicedo, J. C., Singh, S., Carpenter, A. E. & Storkey, A. Automating morphological profiling with generic deep convolutional networks. Preprint at *bioRxiv* 085118 (2016).
26. Doan, M. et al. Deepometry, a framework for applying supervised and weakly supervised deep learning to imaging cytometry. *Nat. Protoc.* **16**, 3572–3595 (2021).
27. Goyal, P. et al. Self-supervised pretraining of visual features in the wild. Preprint at arXiv:2103.01988 (2021).
28. Holmberg, O. G. et al. Self-supervised retinal thickness prediction enables deep learning from unlabelled data to boost classification of diabetic retinopathy. *Nat. Mach. Intell.* **2**, 719–726 (2020).
29. Hadsell, R. et al. Learning long-range vision for autonomous off-road driving. *J. Field Robotics* **26**, 120–144 (2009).
30. Batson, J. & Royer, L. Noise2self: blind denoising by self-supervision. In *Proc. International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) 524–533 (PMLR, 2019).
31. Kobayashi, H. et al. Intelligent whole-blood imaging flow cytometry for simple, rapid, and cost-effective drug-susceptibility testing of leukemia. *Lab. Chip* **19**, 2688–2698 (2019).
32. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *Proc. International Conference on Machine Learning* (eds III Hal, D. & Singh, A.) 1597–1607 (PMLR, 2020).
33. Kolesnikov, A., Zhai, X. & Beyer, L. Revisiting self-supervised visual representation learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 1920–1929 (IEEE, 2019).
34. Deng, J. et al. Imagenet: a large-scale hierarchical image database. In *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255 (IEEE, 2009).
35. Van Den Oord, A., Vinyals, O. et al. Neural discrete representation learning. In *Proc. Advances in Neural Information Processing Systems* (eds Guyon, I. et al.) 6306–6315 (2017).
36. Razavi, A., van den Oord, A. & Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. In *Proc. Advances in Neural Information Processing Systems* (eds Wallach, H. et al.) 14866–14876 (2019).
37. Wu, H. & Flierl, M. Vector quantization-based regularization for autoencoders. In *Proc. AAAI Conference on Artificial Intelligence* vol. **34**, 6380–6387 (AAAI, 2020).
38. Giurgiu, M. et al. Corum: the comprehensive resource of mammalian protein complexes-2019. *Nucleic Acids Res.* **47**, D559–D563 (2019).
39. Donovan-Maiye, R. M. et al. A deep generative model of 3D single-cell organization. *PLoS Comput. Biol.* **18**, e1009155 (2022).
40. Consortium, T. U. Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Res.* **49**, D480–D489 (2021).
41. Schröder, B. A., Wrocklage, C., Hasilik, A. & Saftig, P. The proteome of lysosomes. *Proteomics* **10**, 4053–4076 (2010).
42. Gosney, J. A., Wilkey, D. W., Merchant, M. L. & Ceresa, B. P. Proteomics reveals novel protein associations with early endosomes in an epidermal growth factor–dependent manner. *J. Biol. Chem.* **293**, 5895–5908 (2018).

43. Cheng, Y. & Church, G. M. Biclustering of expression data. In *Proc. International Conference on Intelligent Systems for Molecular Biology* Vol. 8, 93–103 (AAAI Press, 2000).
44. Gerbin, K. A. et al. Cell states beyond transcriptomics: integrating structural organization and gene expression in hIPSC-derived cardiomyocytes. *Cell Syst.* **12**, 670–687 (2021).
45. Viana, M. P. et al. Robust integrated intracellular organization of the human IPS cell: where, how much, and how variable. Preprint at *bioRxiv* 2020-12 (2021).
46. Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. *IEEE Intell. Syst.* **24**, 8–12 (2009).

Methods

Fluorescence image dataset. All experimental and imaging details can be found in our companion study[10]. Briefly, HEK293T cells were genetically tagged with split-fluorescent proteins using CRISPR-based techniques[47]. After nuclear staining with Hoechst 33342, live cells were imaged with a spinning-disk confocal microscope (Andor Dragonfly). Typically, 18 FOV were acquired for each one of the 1,311 tagged proteins, for a total of 24,382 three-dimensional images of dimension $1{,}024\times1{,}024\times22$ voxels.

Image data preprocessing. Each 3D confocal image was first reduced to two dimensions using a maximum-intensity projection along the z axis followed by downsampling in the xy dimensions by a factor of two to obtain a single 2D image per FOV (512×512 pixels). To help our model make use of the nuclear fiducial label, we applied a distance transform to a nucleus segmentation mask (below). The distance transform is constructed so that pixels within the nucleus were assigned a positive value that represents the shortest distance from the pixel to the nuclear boundary, and pixel values outside the nucleus were assigned a negative value that represents the shortest distance to the nuclear boundary (Fig. 1a). For each dual-channel and full FOV image, multiple regions of dimension 100×100 pixels were computationally chosen so that at least one cell is present and centered, resulting in a total of 1,100,253 cropped images. Cells (and their nuclei) that are too close to image edges are ignored. The raw pixel intensities in the fluorescence channel are normalized between 0 and 1, and the nuclear distance channel is normalized between −1 and 1.

Nucleus segmentation. Nuclei are segmented by first thresholding the nucleus channel (Hoechst staining) and then applying a custom algorithm to segment any under-segmented nuclei. In the thresholding step, a background mask is generated by applying a low-pass Gaussian filter to the raw image, then thresholding it using a threshold value calculated by the iterative Minimum Cross Entropy method[48,49]. Under-segmented nuclei in the resulting mask are then segmented by applying the following steps: (1) we generate a second mask by applying a Laplacian of the Gaussian (LoG) filter to the original image, thresholding it at zero, and multiplying it by the background mask from the intensity thresholding step, (2) we morphologically close this second mask and fill holes to eliminate intra-nuclear holes or gaps (empirically, this requires a closing disk of radius at least 4 pixels), (3) we multiply the second mask again by the background mask to restore any true morphological holes that were present in the background mask, (4) we generate a mask of the local minima in the original LoG-filtered image using an empirically selected percentile threshold, and finally (5) we iterate over regions in this local-minima mask and remove them from the second mask if they partially overlap with the background of the refined mask. The second mask is then the final nucleus segmentation mask.

Detailed model architecture. All details of our model architecture are given in Supplementary File 1 and a diagram is shown in Fig. 1b. First, the input image ($100\times100\times2$ pixels) is fed to encoder1 to produce a set of latent vectors that have two destinations: encoder2 and VQ1 VectorQuantizer layer. In the encoder2, higher level representations are distilled from these latent vectors and passed to the output. The output of encoder2 is quantized in the VQ2 VectorQuantizer layer to form what we call 'global representation'. The global representation is then passed to the fc2 classifier for purposes of the classification pretext task. It is also passed on to decoder2 to reconstruct the input data of encoder2. In this way, encoder2 and decoder2 form an independent autoencoder. The function of layer mselyr1 is to adapt the output of decoder2 to match the dimensions of the output of encoder1, which is identical to the dimensions of the input of encoder2. In the case of the VQ1 VectorQuantizer layer, vectors are quantized to form what we call the local representations. The local representation is then passed to the fc1 classifier for purposes of the classification pretext task, as well as concatenated to the global representation that is resized to match the local representations' dimensions. The concatenated result is then passed to the decoder1 to reconstruct the input image. Here, encoder1 and decoder1 form another autoencoder.

Split quantization. In the case of our global representation, we observed that the high level of spatial pooling required (4×4 pixels) led to codebook under-use because the quantized vectors are too few and each one of them has too many dimensions (Fig. 1b). To solve this challenge, we introduced the concept of split quantization. Instead of quantizing all the dimensions of a vector at once, we first split the vectors into subvectors of equal length and then quantize each subvectors using a shared codebook. The main advantage of split quantization when applied to the VQ-VAE architecture is that one may vary the degree of spatial pooling without changing the total number of quantized vectors per representation. In practice, to maintain the number of quantized vectors while increasing spatial pooling, we simply split along the channel dimension. We observed that the global representations' perplexity, which indicates the level of use of the codebook, substantially increases when split quantization is used compared to standard quantization (Fig. 1c). As shown in Supplementary Fig. 1, split quantization is performed along the channel dimension by splitting each channel-wise vector into nine parts, and quantizing each of the resulting 'subvectors' against the same codebook. Split quantization is only needed for the global representation.

Global and local representations. The dimensions of the global and local representations are $4\times4\times576$ and $25\times25\times64$ voxels, respectively. These two representations are quantized with two separate codebooks consisting of 2,048 64-dimensional features (or codes).

Identification pretext task. The part of our model that is tasked with identifying a held-back protein is implemented as a two-layer perceptron built by alternatively stacking fully connected layers with 1,000 hidden units and nonlinear ReLU layers. The output of the classifier is a one-hot encoded vector for which each coordinate corresponds to one of the 1,311 proteins. We use categorical cross entropy as classification loss during training.

Computational efficiency. Due to the large size of our image data (1,100,253 cropped images of dimensions $100\times100\times2$ pixels) we recognized the need to make our architecture more efficient and thus allow for more design iterations. We opted to implement the encoder using principles from the EfficientNet architecture to increase computational efficiency without losing learning capacity[50]. Specifically, we split the model of EfficientNetB0 into two parts to make the two encoders in our model (Supplementary File 1). While we did not notice a loss of performance for the encoder, EfficientNet did not perform as well for decoding. Therefore, we opted to keep a standard architecture based on a stack of residual blocks for the decoder[51]

Training protocol. The whole dataset (1,100,253 cropped images) was split into 8:1:1 into training, validation and testing data, respectively. All results shown in the figures are from testing data. We used the Adam optimizer with the initial learning rate of 0.0004. The learning rate was multiplied by 0.1 every time the validation loss did not improve for four epochs, and the training was terminated when the validation loss did not improve for more than 12 consecutive epochs. Images were augmented by random rotation and flipping in the training phase.

Dimensionality reduction and clustering. Dimensionality reduction is performed using the UMAP[52] algorithm. We used the reference open-source python package umap-learn (v.0.5.0) with default values for all parameters (that is, the Euclidean distance metric, 15 nearest neighbors and a minimal distance of 0.1). We used AlignedUMAP for the clustering performance evaluation to facilitate the comparison of the different projections derived from three variants of the previously described cell-inpainting model[11] (Extended Data Figs. 1 and 2) or all seven variants of our model (Extended Data Figs. 3 and 4). Hierarchical biclustering was performed using seaborn (v.0.11.1) with its default settings.

Ground-truth labels for localization. To evaluate the clustering performance, we used two sets of ground-truth labels at two different cellular scales: a manually curated list of proteins with exclusive organelle-level localization patterns (Supplementary File 2) and 38 protein complexes collected from the CORUM database[38] (Supplementary File 3). The 38 protein complexes were collected based on the following conditions: (1) all subunits are present in the OpenCell data, (2) no overlapping subunit across the complexes and (3) each protein complex consists of more than one distinct subunit.

For the evaluation of feature spectra, we simply extracted the proteins with single-localization annotation based on the localization annotation given by the OpenCell database (Supplementary File 4).

Clustering score. To calculate a clustering score, we assume a collection of n points (vectors) in $\mathbb{R}^m$, $S = \{\mathbf{x}_i \in \mathbb{R}^m | 0 \leq i \leq n\}$, and that we have a (ground truth) assignment of each point $\mathbf{x}_i$ to a class C_j, and these classes form a partition of S:

$$S = \bigcup_j C_j$$

Ideally, the vectors $\mathbf{x}_i$ are such that all points in a class are tightly grouped together, and that the centroids of each class are as far apart from each other as possible. This intuition is captured in the following definition of our clustering score:

$$\Gamma(C_i) = \frac{\sigma^*(\{\mu^*(C_j)\}_j)}{\mu^*(\{\sigma^*(C_j)\}_j)}$$

where $\{.\}_k$ denotes the set of values obtained by evaluating the expression for each value of parameter k, and where μ^* and σ^* stand for the robust mean (median) and robust standard deviation (computed using medians). Variance statistics were obtained by training the model variant five times followed by computing the UMAP ten times per trained model.

Feature spectrum. Extended Data Fig. 6a illustrates the workflow for constructing the feature spectra. Specifically, we first obtain the indices of quantized vectors in the latent representation for each image crop, and then calculate the histogram of indices in all images of each protein. As a result, we obtain a matrix of histograms in which rows correspond to protein identification (ID) and columns to the feature indices (Extended Data Fig. 6b). At this point, the order of the columns (that is, the feature indices) is arbitrary. Yet, different features might be highly correlated and thus either related or even redundant (depending on how 'saturated' the codebook is). To meaningfully order the feature indices, we compute the Pearson correlation coefficient between the feature index 'profiles' (the columns of the matrix) for each pair of feature indices to obtain a 2,048 × 2,048 pairwise correlation matrix (Extended Data Fig. 6c). Next, we perform hierarchical biclustering in which the feature indices with the most similar profiles are iteratively merged[53]. The result is that features that have similar profiles are grouped together (Extended Data Fig. 6d). This ordering yields a more meaningful and interpretable view of the whole spectrum of feature indices. We identified several clusters from the top levels of the feature hierarchy and manually segment them into 11 major feature clusters (ordered i to xi). Finally, for a given protein, we can produce a interpretable feature spectrum by ordering the horizontal axis of the quantized vectors histogram in the same way.

Training cell-inpainting model on OpenCell data. The cell-inpainting model was constructed using the code provided by its original authors (https://github.com/alexxijielu/paired_cell_inpainting). The whole dataset was split into training, validation and testing sets (8:1:1). All results shown in the figures are computed on the basis of the test set. We used the Adam optimizer with the initial learning rate of 0.0004. The learning rate was multiplied by 0.1 every time the validation loss did not improve for four epochs, and the training was terminated when the validation loss did not improve for more than 12 consecutive epochs. The features to generate UMAP were extracted from layers denoted as 'conv3_1', 'conv4_1' and 'conv5_1' by the authors.

Applying cytoself on the Allen Institute dataset. Image data from the Allen Institute were downloaded from https://www.allencell.org/data-downloading.html#DownloadImageData. Patches were made following the same procedure as for the OpenCell dataset including max-intensity projection and downsampling to match pixel resolutions. Nuclear centers were determined using the nuclear label included in the Allen Institute dataset. We randomly selected 80 patches per protein and used these for analysis.

Feature extraction with CellProfiler. CellProfiler v.4.2.1 was used to extract features from nuclear images (without distance transform) and fluorescence protein images. In the case of cytoself, we computed all features compatible to the data including texture features up to scale 15, for a total of 1,397 features that required 2 days of computation. Only features that did not require object detection were used, including granularity, texture and the correlations between the two channels. Each feature was standardized by subtracting its mean followed by dividing by its standard deviation.

Evaluation of feature correlation against protein complex. The Pearson correlation between any two proteins found in both the OpenCell and CORUM databases were computed with their feature spectra as the proximity metrics in the feature space. For each protein, we found the 'nearest protein' with which it had the highest correlation, and incremented the number if the correlation was higher than a given threshold, and if both of them shared at least one complex in the CORUM database. To take into account the strength of correlation, we varied the minimal correlation threshold thus obtaining the curve shown in Supplementary Fig. 6b.

Statistics and reproducibility. All box plots were generated using matplotlib (v.3.4.2). Each box indicates the extent from the first to the third quartile of the data, with a line representing the median. The whiskers indicate 1.5 times the interquartile range. Scipy (v.1.8.0) was used to compute P values and Pearson's correlations.

Software and hardware. All deep-learning architectures were implemented in TensorFlow v.1.15 (ref. [54]) on Python v.3.7. Training was performed on NVIDIA V100-32GB GPUs.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The image data used in this work are available at https://github.com/royerlab/cytoself. The CORUM database is available at http://mips.helmholtz-muenchen.de/corum/. Image data from the Allen Institute are available at https://www.allencell.org/data-downloading.html#DownloadImageData.

Code availability

Source code for the models used in this work is available at https://github.com/royerlab/cytoself.

References

47. Leonetti, M. D., Sekine, S., Kamiyama, D., Weissman, J. S. & Huang, B. A scalable strategy for high-throughput GFP tagging of endogenous human proteins. *Proc. Natl Acad. Sci. USA* **113**, E3501–E3508 (2016).
48. Li, C. H. & Lee, C. Minimum cross entropy thresholding. *Pattern Recog.* **26**, 617–625 (1993).
49. Li, C. & Tam, P. K.-S. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recog. Lett.* **19**, 771–776 (1998).
50. Tan, M. & Le, Q. Efficientnet: rethinking model scaling for convolutional neural networks. In *Proc. International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) 6105–6114 (PMLR 2019).
51. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
52. McInnes, L., Healy, J. & Melville, J. UMAP: uniform manifold approximation and projection for dimension reduction. Preprint at *arXiv* arXiv:1802.03426 (2018).
53. Rokach, L. & Maimon, O. (eds) *Data Mining and Knowledge Discovery Handbook* 321–352 (Springer, 2005).
54. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous systems. *tensorflow.org* https://www.tensorflow.org/ (2015).

Acknowledgements

We thank our colleagues at the Chan Zuckerberg Biohub, S. Schmid, M. Bucci, A.C. Solak, B. Yang, M. Lange, S. Vijaykumar, L. Hyman and M. Hein, for insightful discussions, feedback and for reviewing the manuscript. We thank K. Kim for assistance with data analysis. We thank our colleagues M. Wu and J. Zou from Stanford University for advice. We thank A. Lakoduk, J. Bragantini and S. Schmid for reviewing the manuscript and A. C. Solak for helping with coding. Finally, we thank the Japan Society for the Promotion of Science and its overseas research fellowships and the Chan Zuckerberg Biohub and its donors for funding this work.

Author contributions

H.K., M.D.L. and L.A.R. conceived the piece. H.K. and K.C.C. performed data analysis. All the authors wrote the manuscript and designed the figures.

Competing interests

The authors declare no competing interests.

Additional information

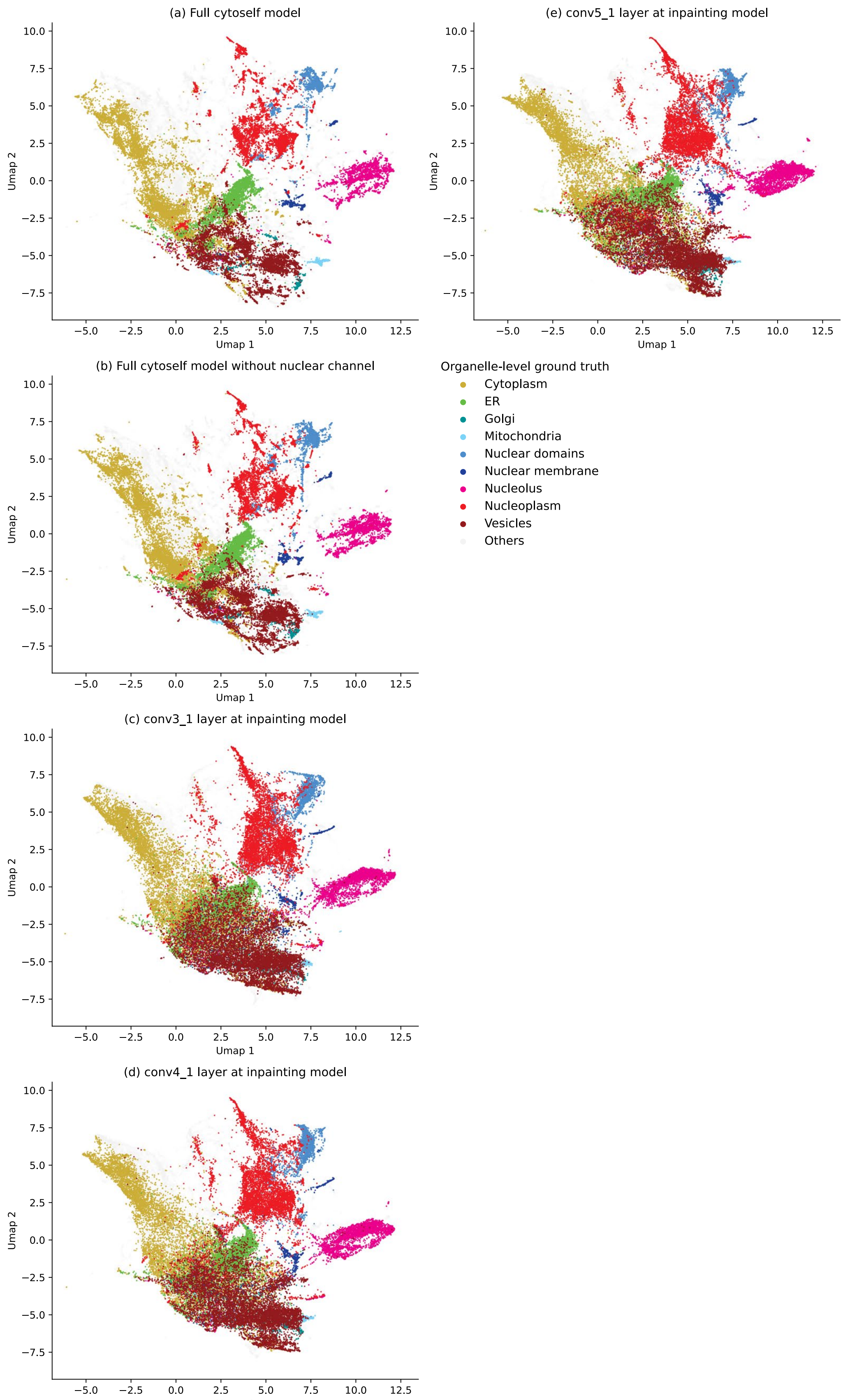
Extended data are available for this paper at https://doi.org/10.1038/s41592-022-01541-z.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-022-01541-z.

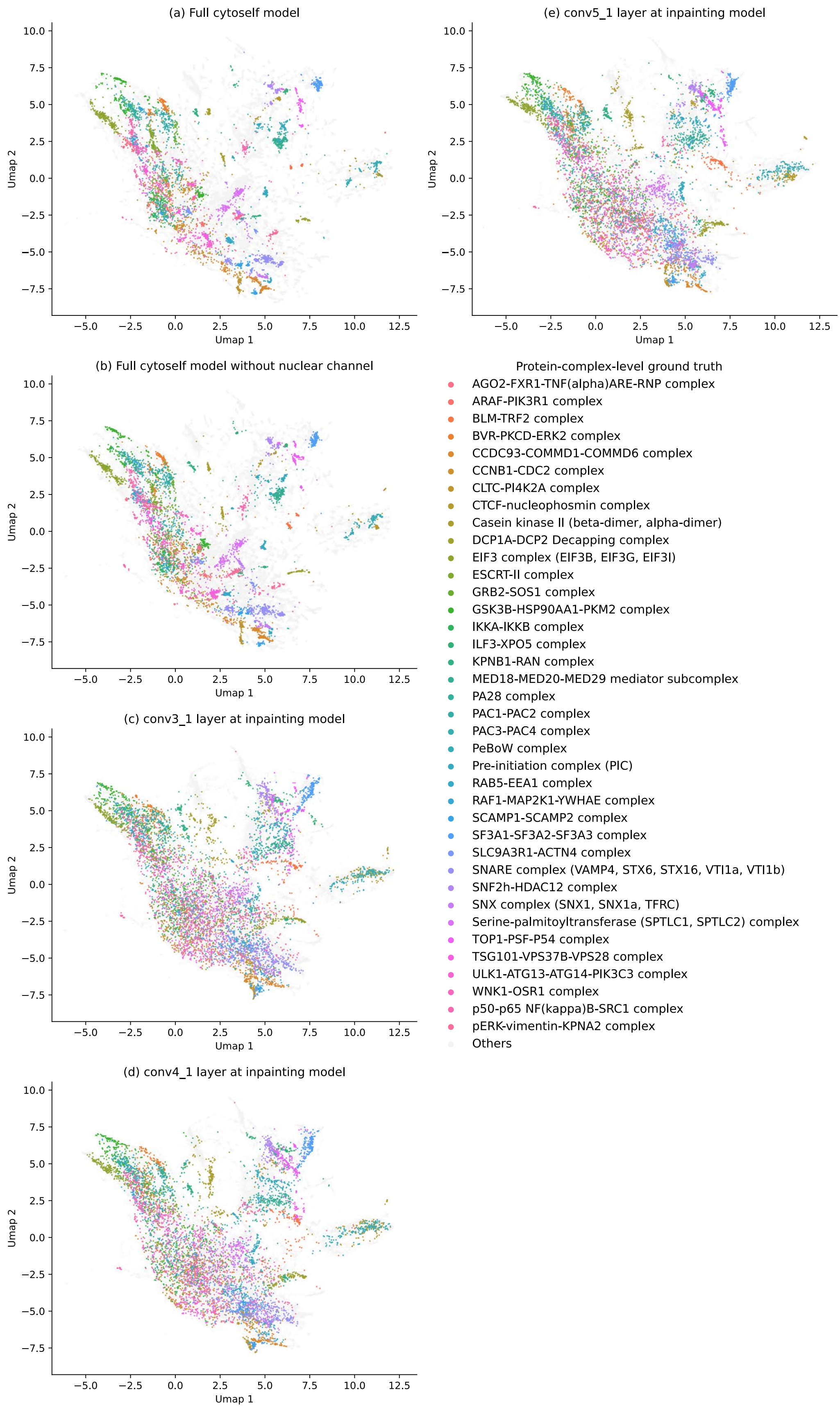
Correspondence and requests for materials should be addressed to Hirofumi Kobayashi, Manuel D. Leonetti or Loic A. Royer.

Peer review information *Nature Methods* thanks Assaf Zaritsky and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Rita Strack, in collaboration with the *Nature Methods* team.

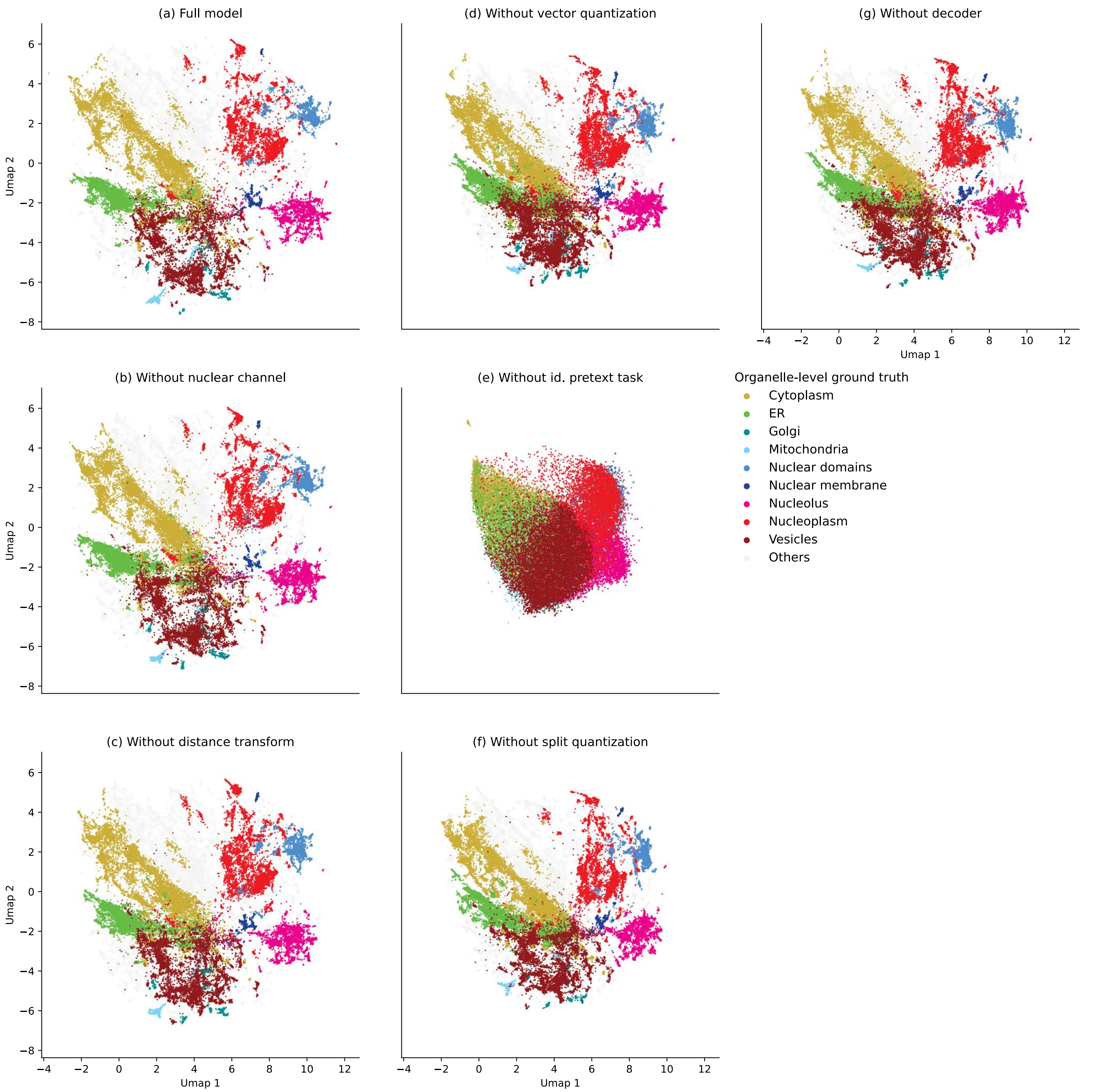
Reprints and permissions information is available at www.nature.com/reprints.



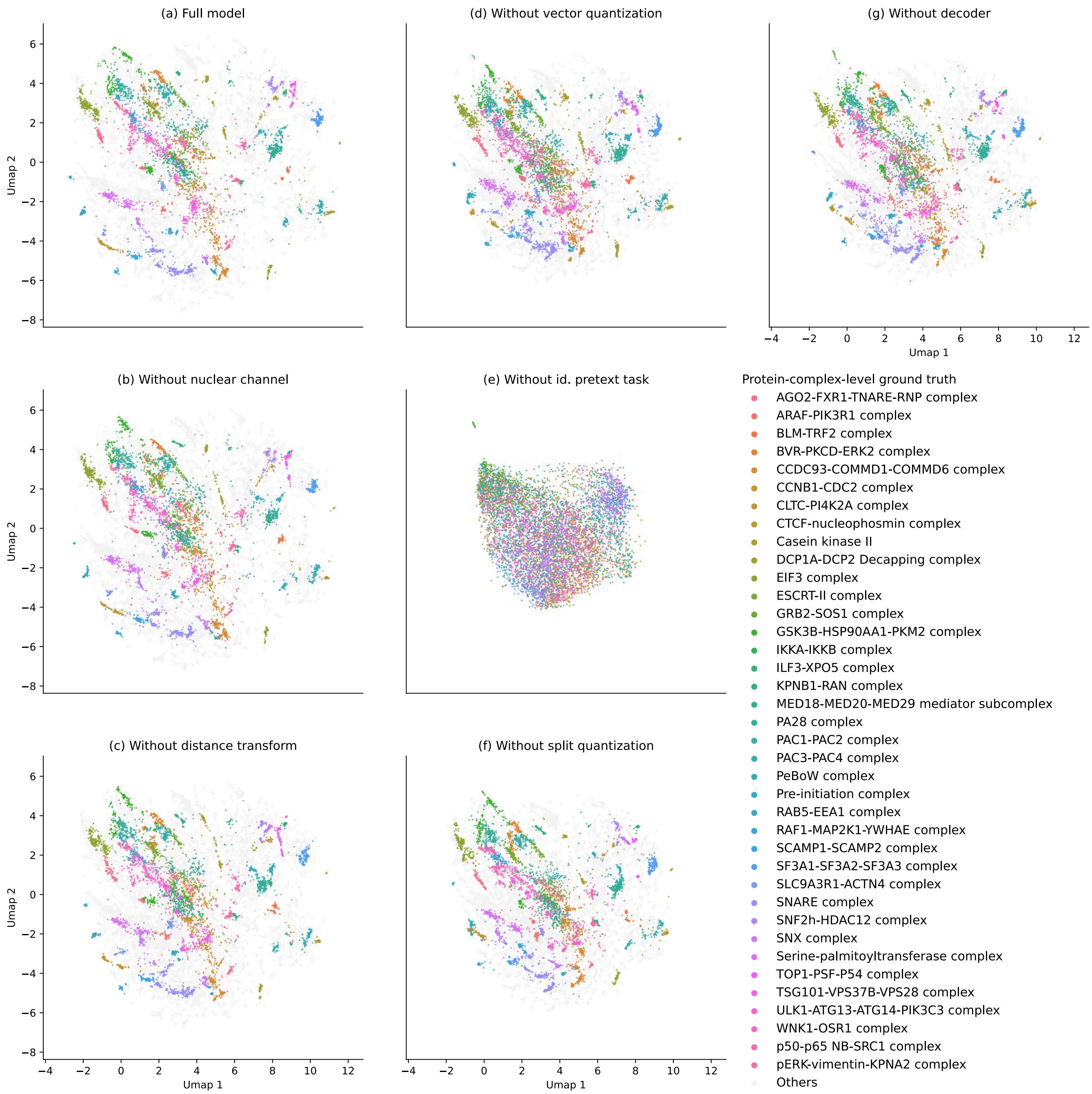
Extended Data Fig. 1 | Comparing the UMAP representations between *cytoself* and cell-inpainting annotated with organelle-level ground truth. Aligned UMAPs are given to aid visual comparison.



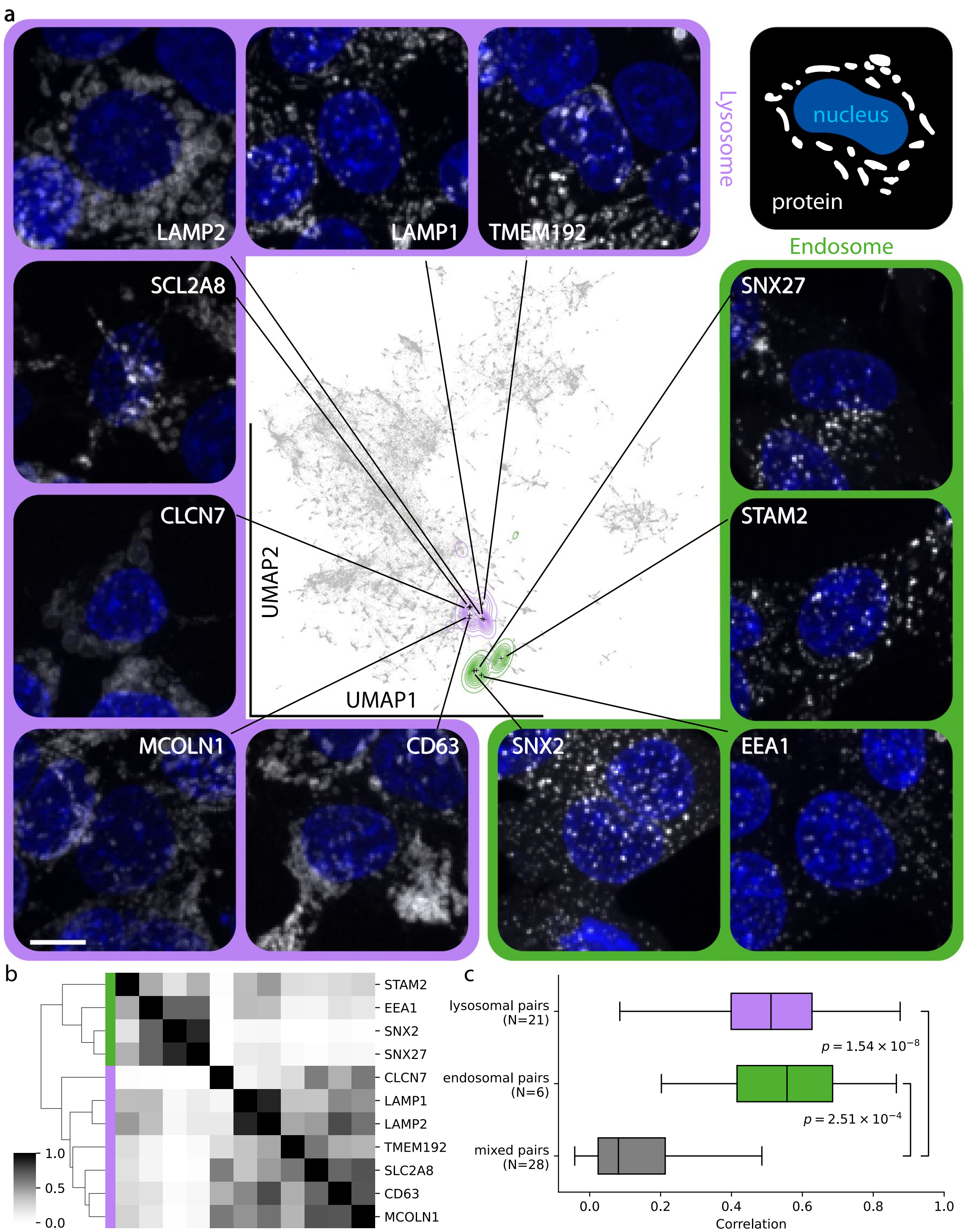
Extended Data Fig. 2 | Comparing the UMAP representations between *cytoself* and cell-inpainting annotated with protein-complex-level ground truth. Aligned UMAPs are given to aid visual comparison.



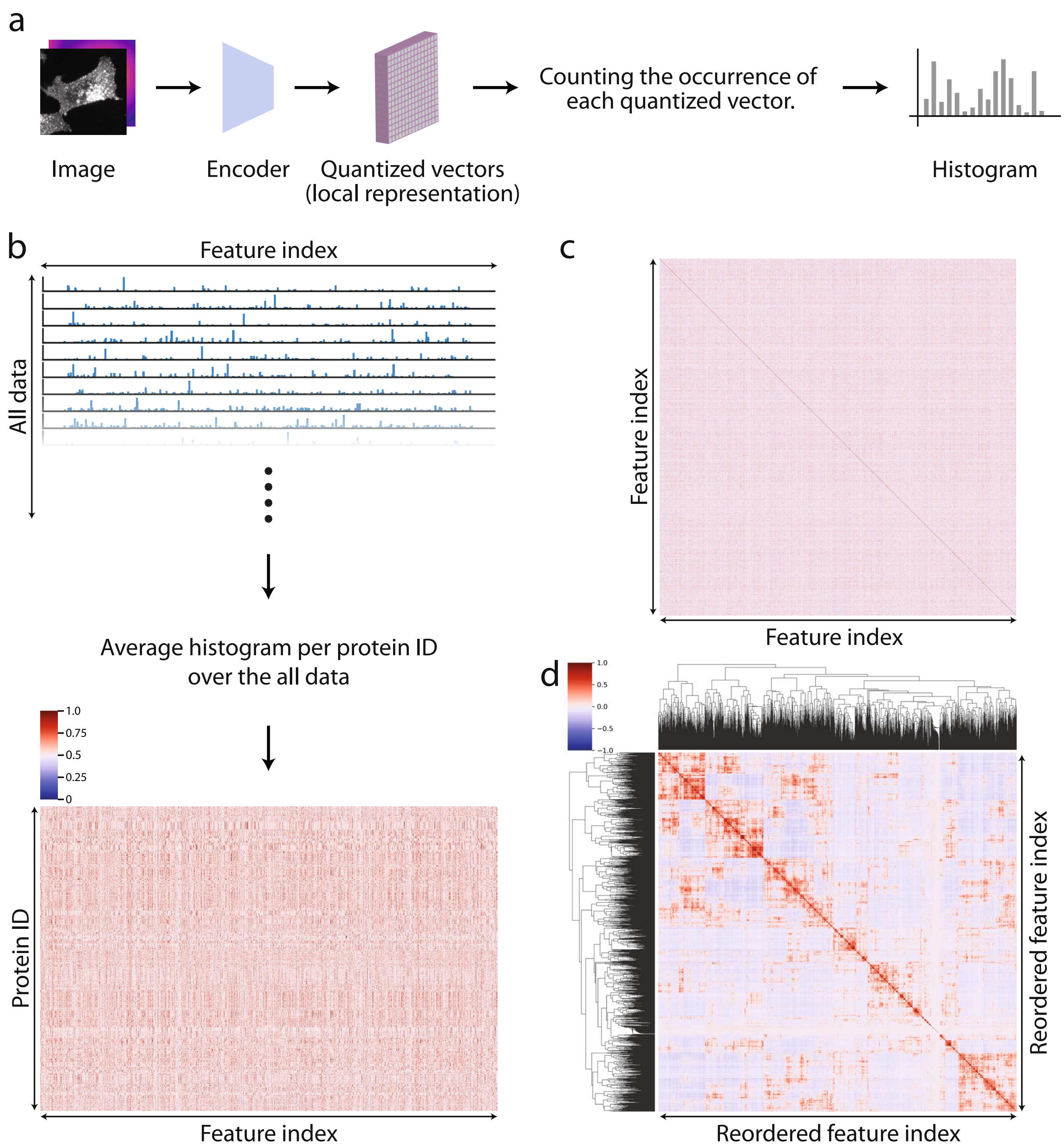
Extended Data Fig. 3 | Identifying the essential components of our model with organelle-level ground truth. Protein localization UMAPs are derived after removing each components of our model separately. Aligned UMAPs are given to aid visual comparison.



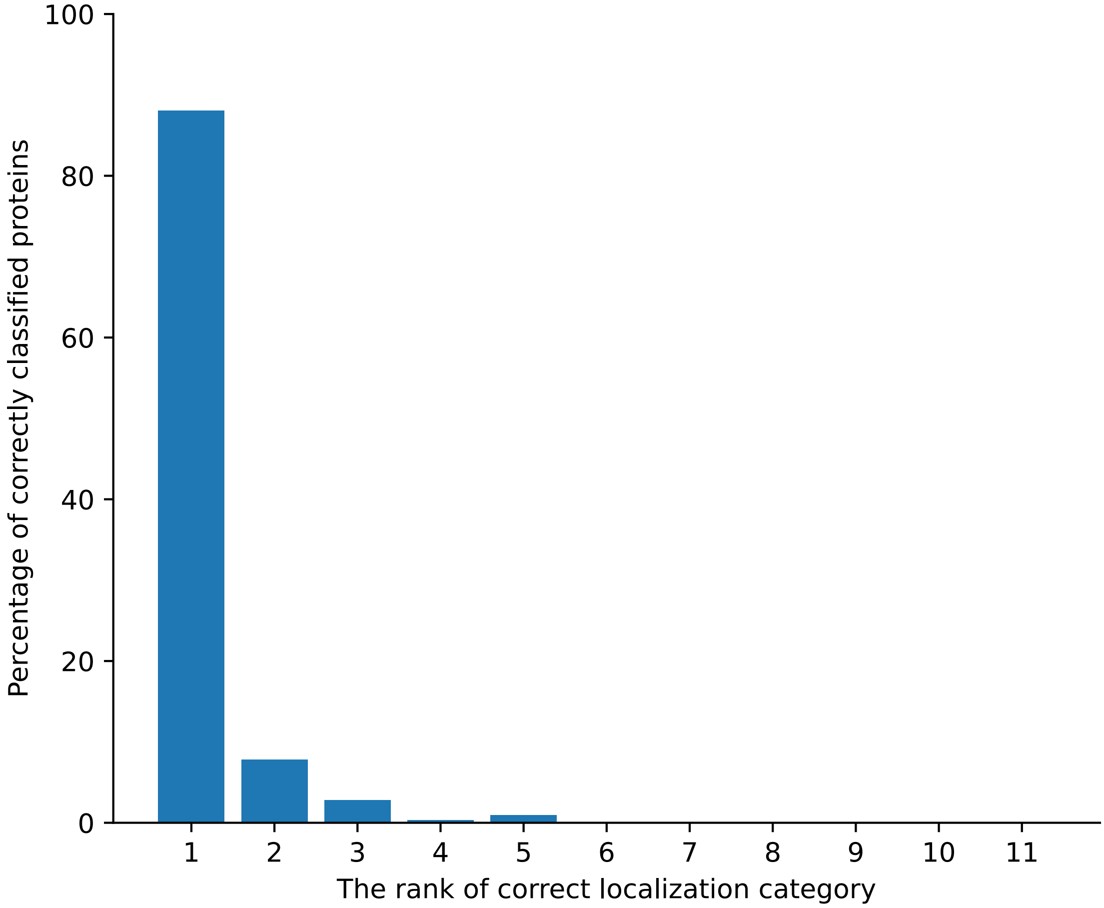
Extended Data Fig. 4 | Identifying the essential components of our model with protein-complex-level ground truth. Protein localization UMAPs are derived after removing each components of our model separately. Aligned UMAPs are given to aid visual comparison.



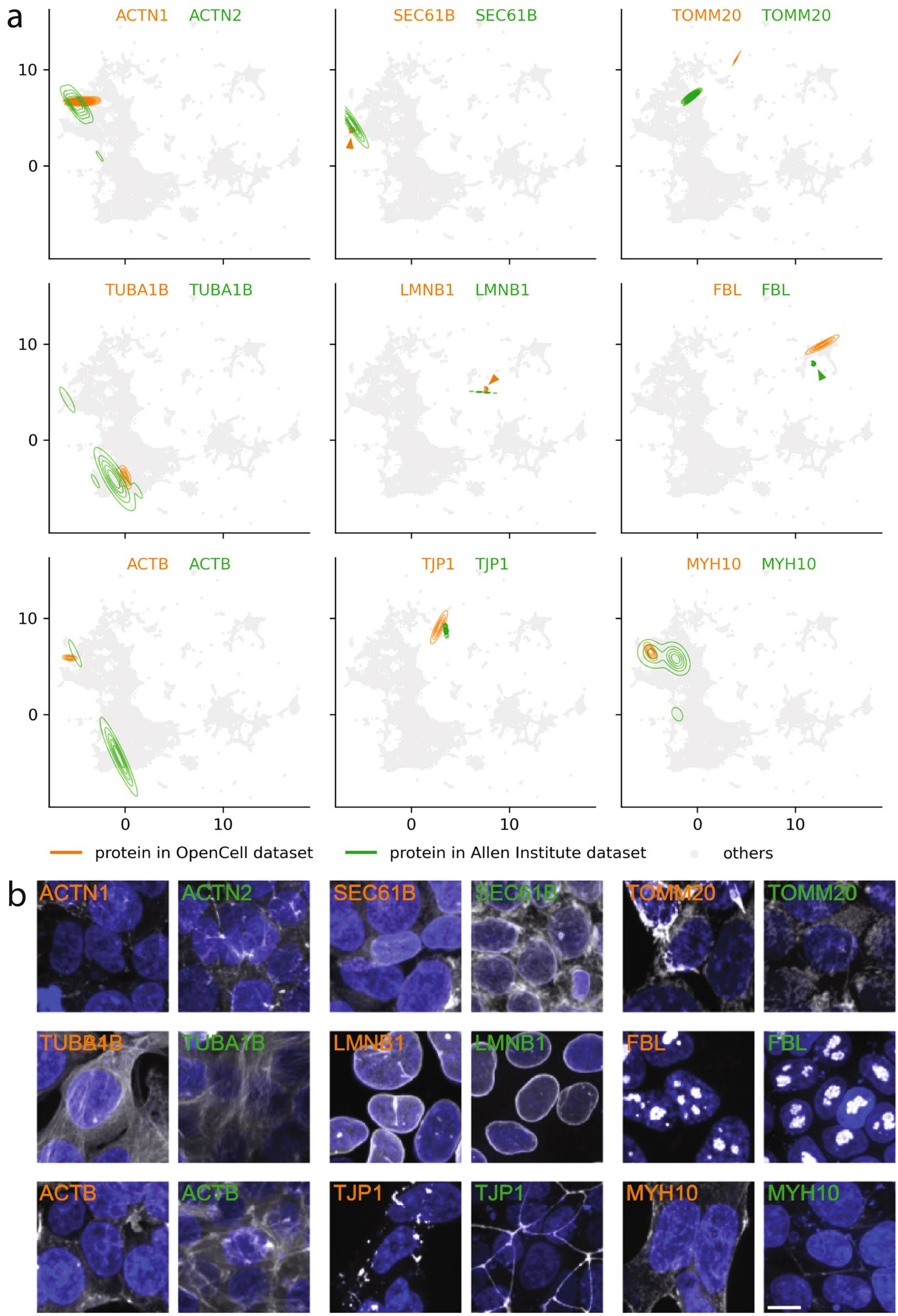
Extended Data Fig. 5 | cytoself discriminates between lysosomal and endosomal proteins. (a) We selected 11 proteins in OpenCell annotated in Uniprot as lysosomal or endosomal that are independently confirmed as such by mass spectrometry[41,42]. We show that *cytoself* is able to distinguish the lysosomal from endosomal proteins solely on the basis of the fluorescence images. All of these proteins are annotated on the basis of the images as 'vesicles' in both HPA and OpenCell. The min and max intensities of each image are adjusted to ensure comparable visibility. All representative images were randomly selected from each protein. Scale bar: 10 μm. **(b)** Clustering of these proteins on the basis of the feature spectra. **(c)** Feature spectra correlations for pairs of lysosomal and endosomal proteins, and for mixed lysosomal-endosomal pairs. Each box indicates the extent from the first to the third quartile of the data, with a line representing the median. The whiskers indicates 1.5 times the inter-quartile range. The p-values are computed using two-sided Mann–Whitney U test.



Extended Data Fig. 6 | Process of constructing feature spectra. (a) First, the quantized vectors in the local representation were extracted and converted to a histogram by counting the occurrence of each quantized vector. **(b)** Next, taking the average of the histograms per protein ID over all the data to create a 2D histogram. **(c)** Pearson's correlations between any two representation indices were calculated and plotted as a 2D matrix. **(d)** Finally, hierarchical clustering was performed on the correlation map so that similar features are clustered together, revealing the structure inside the local representation. The whole process corresponds to the Spectrum Conversion in Fig. 1a.

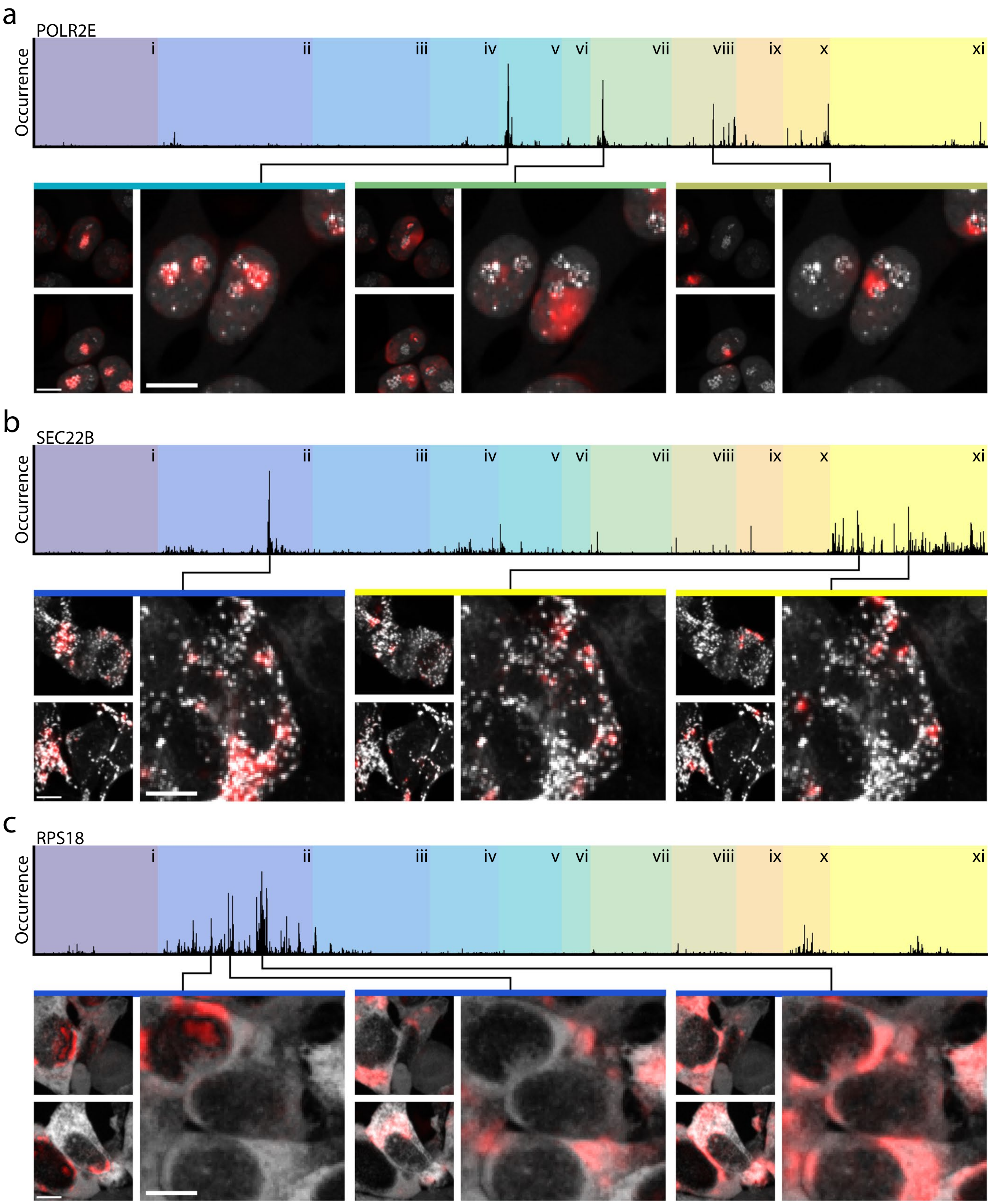


Extended Data Fig. 7 | Predicting the localization category of mono-localized OpenCell proteins by correlating the *cytoself* spectra of each protein with the representative spectra of each category – in a leave-one-out fashion. Result: 88% of proteins are correctly classified. For 96% of proteins the correct annotation is within the top 2 predictions, and for 99% it is within the top 3 predictions.



Extended Data Fig. 8 | See next page for caption.

Extended Data Fig. 8 | Visualizing the predicted localization categories of proteins present both in OpenCell and the Allen Institute dataset. The *cytoself* model is trained only on OpenCell data which is the same *full* model used throughout this work. UMAPs **(a)** and example images **(b)** from OpenCell and Allen Institute datasets for the same or related proteins. The min and max intensities of each image are adjusted to ensure comparable visibility. All representative images were randomly selected. Scale bar 10 μm. Protein names and contour lines in orange color are from OpenCell dataset, and those in green color are from Allen Institute dataset.



Extended Data Fig. 9 | Interpreting image spectral features. Feature spectra were computed for each example proteins (a) POLR2E, (b) SEC22B, and (c) RPS18. Subsequently, information derived from the indicated major peaks of their feature spectra was removed by zeroing them out before passing the features again through the decoder. Highlighted in red are the differences between the resulting output images for the corresponding features and reconstructed image with full features on. The feature classes outlined in Fig. 5 are shown as background color for reference. The pixel intensities are rescaled to the minimum and maximum of each image. Scale bars: 10μm.

nature research

Corresponding author(s): Loic A. Royer, Hirofumi Kobayashi, Manuel Leonetti

Last updated by author(s): Apr 27, 2022

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's d, Pearson's r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Data per se was not collected in this study.
Data analysis	The source code is available at https://github.com/royerlab/cytoself

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The image data used in this work is available at https://github.com/royerlab/cytoself.
The CORUM database is independently available at: http://mips.helmholtz-muenchen.de/corum/
Image data from the Allen Institute is independently available at: https://www.allencell.org/data-downloading.html#DownloadImageData

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We used all available data from OpenCell database, in which 1311 proteins were tagged and 24,382 fluorescence images were acquired.
Data exclusions	No data was excluded.
Replication	All the deep learning models were trained for 5 times. All the UMAPs were computed for 10 times.
Randomization	The randomization happens internally when training a deep learning model or computing UMAP. For other analysis, we used all available data and did not need randomization.
Blinding	We trained the cytoself model by holding out one protein.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study	
☒	☐	Antibodies
☐	☒	Eukaryotic cell lines
☒	☐	Palaeontology and archaeology
☒	☐	Animals and other organisms
☒	☐	Human research participants
☒	☐	Clinical data
☒	☐	Dual use research of concern

Methods

n/a	Involved in the study	
☒	☐	ChIP-seq
☒	☐	Flow cytometry
☒	☐	MRI-based neuroimaging

Eukaryotic cell lines

Policy information about cell lines

Cell line source(s)	ATCC CRL-3216
Authentication	Short Tandem Repeat Analysis (cell culture facility, UC Berkeley)
Mycoplasma contamination	Mycoplasma contamination: Cell lines were tested negative for mycoplasma (MycoAlert Plus, Lonza)
Commonly misidentified lines (See ICLAC register)	None