NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS

as a manuscript

Nikita Moshkov

APPLICATION OF DEEP LEARNING ALGORITHMS TO SINGLE-CELL SEGMENTATION AND PHENOTYPIC PROFILING

PhD Dissertation Summary

for the purpose of obtaining academic degree
Doctor of Philosophy in Computer Science

Academic supervisors:
Attila Kertész-Farkas, Ph.D.
Peter Horvath, Ph.D.
Juan C. Caicedo, Ph.D.

Moscow - 2022

Contents

1 Introduction

1.1 The relevance of research

The decisive element in approaching fundamental questions in biology and designing efficient disease treatments is the understanding of cellular molecular processes [1]. The analysis of the single cell has become one of the most important challenges in natural sciences in the 21st century. The game-changing idea [2] is to treat every single cell in tissues as a separate building block with its state and therefore treat tissues as a diverse set of such building blocks, rather than as a homogeneous entity. The means of an extensive investigation of this idea were the new high-throughput technologies for genome sequencing, proteomics, metabolomics and imaging.

Such advancements made it possible to automatically and objectively analyze even on scales as large as millions and billions of cells, thus we have an opportunity to perform high-throughput experiments with single cells (live-cell imaging [3] [4], gene expression profiling [5] and proteomics [6]) and then perform analysis with computational methods, applicable for the obtained type of data and try to make biological sense out of this data.

Different types of data (or data modalities) can allow us to inspect the state of each particular cell from different perspectives. One of the practical tasks, where all the possible information can be useful to make decisions, is drug discovery, especially in personalized medicine. The biggest challenge is to accurately and cost-effectively combine and use the existing expensive treatment modalities.

Here we focus mostly on the imaging data and one of the first steps of the image-based analysis of single cells is *cell or nucleus segmentation* – classification of each pixel as a background or foreground (semantic segmentation), or determining if the pixel belongs to a specific object (instance segmentation), examples are in Figure 1. In recent years this field has been emerging by adopting and creating deep learning algorithms for this task, bringing significant improvements [7].

The segmentation might be followed by the identification of biological phenotypes through the quantification of cell morphology, variation of which might show, for instance, differences between treated and not treated cells in drug screening experiments [8]. The phenotypes can be described by feature-vectors, also called *profiles* and the process of the extraction is called profiling and morphological profiling is also might be referred to as *image-based profiling* [9] [10].

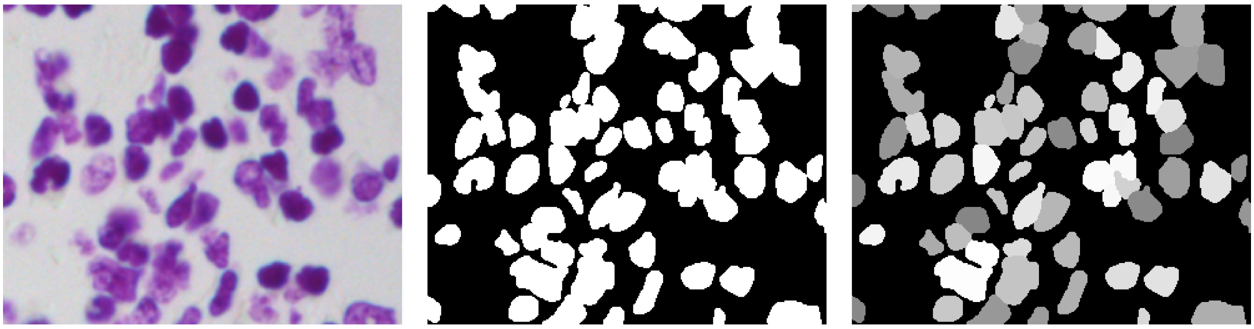


Figure 1: Example of segmentation left-to-right: original image, semantic segmentation, instance segmentation. The source of the image and segmentations: Data Science Bowl 2018 dataset [11].

1.2 Specific aims of the thesis

1.2.1 Review existing methods for cell segmentation

Image pre-processing and nucleus (or cell) segmentation are usually the first steps of the analysis of single-cell images. The accurate segmentation affects the quality of the following downstream analysis, so this step is crucially important.

The author of the thesis contributed to the review paper [7], which puts together the state of the field of nucleus segmentation in 2020-2021. Besides the segmentation methods for 2D and 3D data, it also covers the pre- and post-processing methods, existing datasets and tools for annotation of cellular images.

1.2.2 Deep-learning assisted nuclei annotation

To train a single-cell (nuclei) segmentation based on deep learning, annotated data is needed and the bigger the dataset is, the more robust the model will be. Manual annotation is an expensive process as it requires a significant amount of time and effort from biology experts. To make the annotation process faster and more accurate, a plugin AnnotatorJ [12] for ImageJ/FIJI [13] (the software for bioimage analysis) was developed which combines single-cell identification with deep learning and manual annotation.

1.2.3 Evaluate test-time augmentation approach for nuclei segmentation

Test-time augmentation was an existing approach to improve image classification [14]. In this thesis, test-time augmentation for nuclei segmentation is evaluated. The trained deep learning model for segmentation processes the original input image and several transformed variants of the same image. The obtained segmentation results are then merged. The core idea is that the combination of segmentation results from the original image and its transformed variants will perform better than the segmentation of just the original image or at least will give us hints about uncertain segmentations. The final result is an experimental evaluation of this approach for two popular segmentation deep learning networks.

1.2.4 Image-based morphological profiling with deep learning

The use of deep learning models for image-based profiling (phenotyping of single cells) is investigated. Those deep learning models can be either pre-trained (with ImageNet dataset [15]) or trained (weakly supervised) for the particular single-cell dataset. Using those models, it is possible to extract features (profiles) of the single cells. The obtained features are used in the downstream analysis afterwards (for instance, to predict the mechanisms of action of drugs). We investigate if the features obtained with deep learning networks provide better results in the downstream analysis than classical morphological features [16], particularly for the images obtained with Cell Painting [10] (also see in 2.2).

1.2.5 Assess different sources of features for drug screening

The relative predictive power is compared for three high-throughput sources of features: representations of chemical structures [17] of compounds, gene expression phenotypic profiles obtained with L1000 assay [18] and image-based morphological profiles obtained from Cell Painting [10] images processed with CellProfiler [19] for the task of assay-compound activity prediction.

1.3 Importance of the presented work

The review [7] (Aim 1.2.1) of the most recent 2D and 3D segmentation methods provides insights for practitioners about usage and the most suitable methods for different microscopy modalities. As the end-users of the segmentation pipelines are usually biologists, the guidance for the most effective and easy-to-use framework might be helpful to the community, as accurate segmentation is crucially important for the following downstream tasks.

The usage of deep learning-based algorithms is not possible without accurately annotated image datasets and in the field of nuclei segmentation, such datasets are usually built by experts. We have developed a tool [12] (Aim 1.2.2) to make the creation of annotated nuclei datasets faster, more comfortable and, thus, cheaper.

One of the possible ways to obtain better segmentation is to apply post-processing methods. One of such potential methods is test-time augmentation, which is traditionally used for image classification. The systematic evaluation [20] (Aim 1.2.3) of this method for the task of segmentation of nuclei for the most popular deep learning frameworks and the most popular nuclei dataset so far provides insights into its usefulness.

The main goal of image-based morphological profiling is to get such feature representation that accurately captures the cell state [21]. Deep learning networks for image classification might be able to capture such representations, especially with post-processing steps, such as aggregation. Deep learning image-based morphological profiling combined with a cost-efficient Cell Painting assay [10] can be used in drug discovery and other biologically relevant questions (Aim 1.2.4).

Besides morphology, gene expression profiles and information and representations of chemical structures [17] are useful for extracting useful information in the drug discovery task. The comparison (Aim 1.2.5) of their predictive power can provide insights and demonstrate the usefulness of machine learning models for early-stage drug discovery processes.

1.4 Publications

Papers related to the research topic:

- **Moshkov N.**, Mathe B., Kertesz-Farkas A., Hollandi R., Horvath P. Test-time augmentation for deep learning-based cell segmentation on microscopy images. Scientific Reports. 2020. Vol. 10, 5068. Q1 journal, IF 3.998 (2020). DOI: https://doi.org/10.1038/s41598-020-61808-3

- Hollandi R.*, **Moshkov N.***, Paavolainen L., Tasnadi E., Piccinini F., Horvath P. Nucleus segmentation: towards automated solutions. Trends in Cell Biology. 2022. Q1 journal, IF 20.808 (2021). DOI: https://doi.org/10.1016/j.tcb.2021.12.004

- Hollandi R., Diosdi A., Hollandi G., **Moshkov N.**, Horvath P. AnnotatorJ: an ImageJ plugin to ease hand-annotation of cellular compartments. Molecular Biology of the Cell. 2020 Vol. 31. № 20. P. 2157-2288. Q1 journal, IF 3.791 (2020). DOI: https://doi.org/10.1091/mbc.E20-02-0156

Preprints related to the research project:

- **Nikita Moshkov**, Tim Becker, Kevin Yang, Peter Horvath, Vlado Dancik, Bridget K. Wagner, Paul A. Clemons, Shantanu Singh, Anne E. Carpenter, Juan C. Caicedo. Predicting compound activity from phenotypic profiles and chemical structures bioRxiv 2020.12.15.422887, DOI: https://doi.org/10.1101/2020.12.15.422887

- **Nikita Moshkov**, Michael Bornholdt, Santiago Benoit, Matthew Smith, Claire McQuin, Allen Goodman, Rebecca Senft, Yu Han, Mehrtash Babadi, Peter Horvath, Beth A. Cimini, Anne E. Carpenter, Shantanu Singh, Juan C. Caicedo. Learning representations for image-based profiling of perturbations. bioRxiv 2022.08.12.50378, DOI: https://doi.org/10.1101/2022.08.12.503783

Conferences, related to the research project:

- HEPTECH AIME19 AI & ML (2019). Test-time augmentation for deep learning-based cell segmentation on microscopy images (poster). Link: https://indico.wigner.hu/event/1058/contributions/2542/

Papers unrelated to the research topic published in 2017-2022:

- **Moshkov N.***, Smetanin A.*, Tatarinova T. Local ancestry prediction with PyLAE. PeerJ. 2021. Article 12502. Q2 journal, IF 2.816. DOI: https://doi.org/10.7717/peerj.12502

- Piccini F., Balassa T., Carbonaro A., Diosdi A., Toth T., **Moshkov N.**, Tasnadi E. A., Horvath P. Software tools for 3D nuclei segmentation and quantitative analysis in multicellular aggregates. Computational and Structural Biotechnology Journal. 2020. Vol. 18. P. 1287-1300. IF 6.018 (2020), Q1 journal. DOI: `https://doi.org/10.1016/j.csbj.2020.05.022`

- Grexa I., Diosdi A., Harmati M., Kriston A., **Moshkov N.**, Buzas K., Pietiäinen V., Koos K., Horvath P. SpheroidPicker for automated 3D cell culture manipulation using deep learning. Scientific Reports. 2021. Vol. 11, 14813. Q1 journal, IF 4.379 (2021). DOI: `https://doi.org/10.1038/s41598-021-94217-1`

- Kornienko I. V., Faleeva T. G., Schurr T. G., Aramova O. Y., Ochir-Goryaeva M. A., Batieva E. F., Vdovchenkov E. V., **N. E. Moshkov**, Kukanova V. V., Ivanov I. N., Sidorenko Y. S., Tatarinova T. V. Y-Chromosome Haplogroup Diversity in Khazar Burials from Southern Russia. Russian Journal of Genetics. 2021. Vol. 57. No. 4. P. 477-488. IF 0.581. DOI: https://doi.org/10.1134/S1022795421040049

Conferences, unrelated to the research project:

- HEPTECH AIME ML&VA on Clouds (2018). Image database generation techniques for DIC brain tissue cell segmentation (poster). Link: `https://indico.wigner.hu/event/904/contributions/1874/`

2 Background

2.1 Neural networks for segmentation of nuclei and single cells

The history of automated approaches to segment cells and nuclei starts around 60 years ago and those very first approaches were solely based on intensity thresholding [22]. For a very long time, the intensity thresholding (example in Figure 2) was a dominant approach, being the only part of the segmentation pipelines or combined with other classical approaches. Later on, just before the deep learning era, there were other approaches for nuclei segmentation based on classical machine learning [23], active contours [24] [25] and the multilayer gas of circles model [26]. The complexity of biological questions together with the data to be analyzed (developmental biology [27], drug discovery [28], functional genomics [29] and pathology [30]) have started to demand more accurate cell segmentation, and the field has started to seek general solutions to nuclei segmentation task. The adoption of convolutional neural networks and the availability of computational resources to train convolutional deep learning models allowed us to leap toward such solutions.

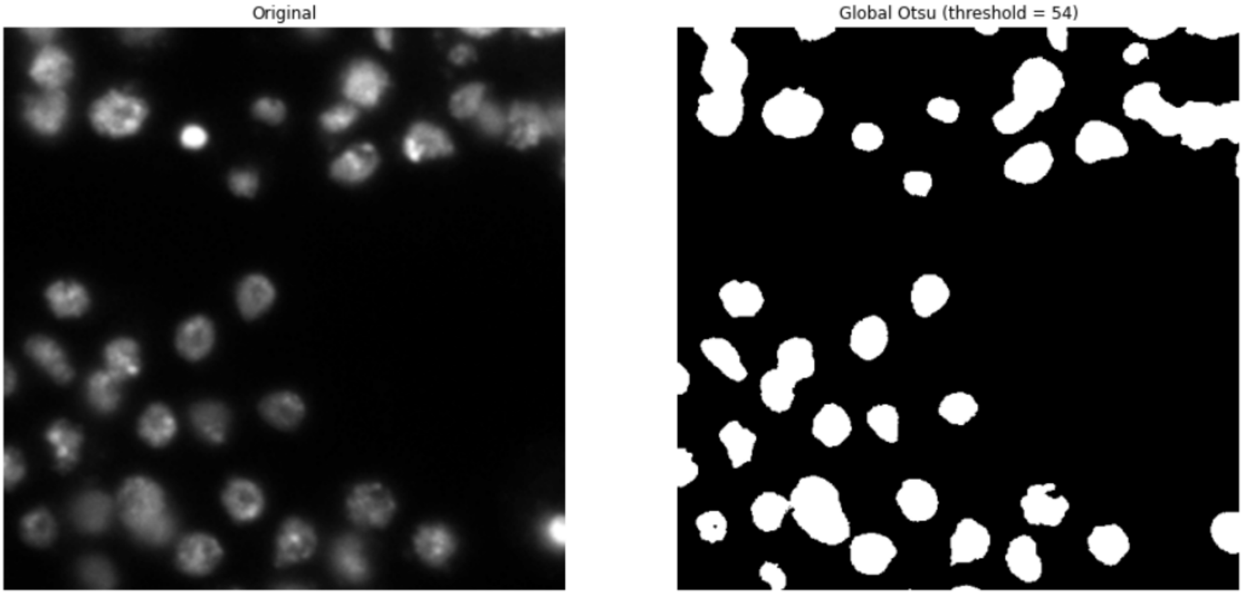


Figure 2: Microscopy image and segmentation mask produced by Otsu thresholding [31] from Scikit-image [32]. The source of the image: Data Science Bowl 2018 dataset [11].

One of the first steps of the image-based analysis of single cells is cell/nuclei detection/segmentation. ***Single-cell segmentation*** (and image segmentation in general) is a vastly developing field: with increased performance of GPUs (graphical processing units) and deep learning neural networks like U-Net [33] (see also 2.1.1), which was the breakthrough for deep learning-based segmentation for biological images (and in the field of deep learning-based segmentation in general). This approach still serves as a baseline for semantic segmentation tasks (i.e. pixel classification) and is used as part of the recent general nucleus/cell segmentation pipelines such as CellPose [34], and StarDist [35] and their derivatives. Besides specialized methods for cell segmentation, methods initially developed for natural image segmentation, like Mask R-CNN [36] (see also 2.1.2) are also applied to single-cell segmentation tasks either by simple fine-tuning or as a part of a complex segmentation pipeline [37].

In addition to deep learning networks themselves, there are common training techniques for regularization, and therefore to train more robust models such as data augmentation for training (modification of original training data by rotating, flipping or adding noise) [38], dropout layers [39], L1 or L2 regularization [40]. Single-cell (nuclei) segmentation task is not an exception to using those techniques.

2.1.1 U-Net

U-Net [33] (Figure 3) is a deep learning-based architecture, developed primarily for biological-image semantic segmentation in 2015 (also was a winner of the ISBI cell tracking challenge). It takes its name from the U-shape encoder-decoder architecture: the input data is firstly compressed by convolutional layers and then expanded back to its original size. U-Net is still widely used as a baseline in nuclei segmentation and there are numerous pipelines based on it for different datasets [7].

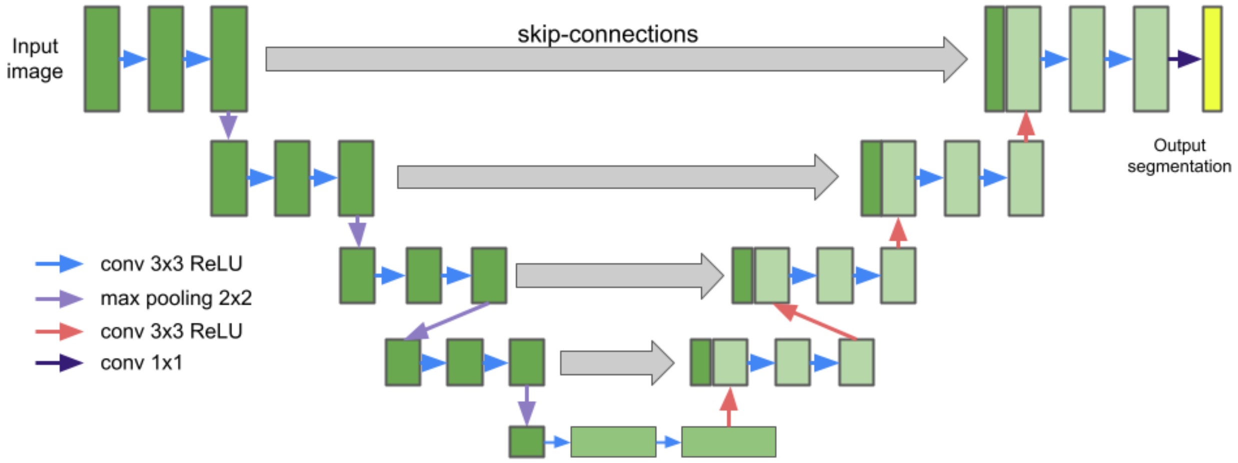


Figure 3: Standard U-Net architecture.

2.1.2 Mask R-CNN

Mask R-CNN [36] (Figure 4) was developed in 2017 for instance segmentation (each pixel in the image is assigned to a separate object) of natural images. Mask R-CNN uses a ResNet [41] architecture as a backbone (usually ResNet50 or ResNet101), which is followed by a region proposal network (RPN). This is stage one of Mask R-CNN, which finishes with a set of proposed regions with objects.

RoIAling (RoI - region of interest) is one of the key enhancements of Mask R-CNN over Faster R-CNN [42], which uses RoI pooling. Both of those operations in principle extract RoIs from the feature maps, RoIAling is more precise. It is followed by the head layers: they predict the class, box offset and an output binary mask for each region of interest (RoI).

Classes are not taken into account for mask generation. RoIAling and the head layers are stage two of Mask R-CNN.

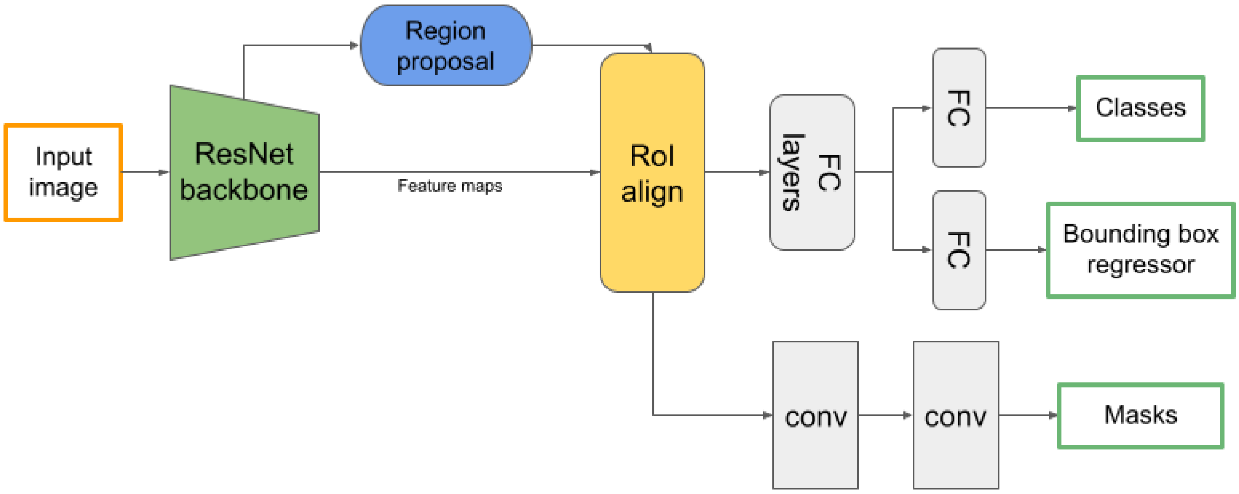


Figure 4: Standard Mask R-CNN architecture.

2.2 Cell Painting and phenotypic profiling

The *target-based approach* used to be dominant in drug discovery, but currently, the *phenotypic approach* to drug discovery takes advantage [43]. Target-based drug discovery focuses on the search for drug targets – gene products, which are the starting point for investigation, and then researchers come up with an idea of how to affect it [44]. The phenotypic approach to drug discovery is empirical: a large number of compounds are tested in target-agnostic assay and the phenotypic variation is monitored [45]. Phenotypic drug discovery expanded the search space of drugs, targets and mechanisms of action making their discovery possible [46].

One way to identify phenotypic variation is through the quantification of cell morphology, which might demonstrate the differences between treated and not treated cells in drug screening experiments [8] [9]. An effective assay for phenotypic-based drug discovery is Cell Painting. This assay was designed to capture as many biologically meaningful morphological features as possible while maintaining the protocol compatible with existing microscopy systems and at the same time keeping it relatively cheap [10]. The output images are five-channel and capture eight cellular compartments (see Figure 5).

Cell morphology might be described by a vector of features - or *profile* (either for individual cells or aggregated for a population of cells), extracted by a multi-stage pipeline [48]. This task can be referred to as ***morphological profiling*** [9] [48] or with broader term ***phenotypic profiling*** [49]. The extracted profiles are processed in downstream analysis of interest. The most popular software to create pipelines to obtain morphological profiles of the cells is CellProfiler [19], the features are hand-crafted, though features obtained with deep learning models are to be researched [50] [51] [52].

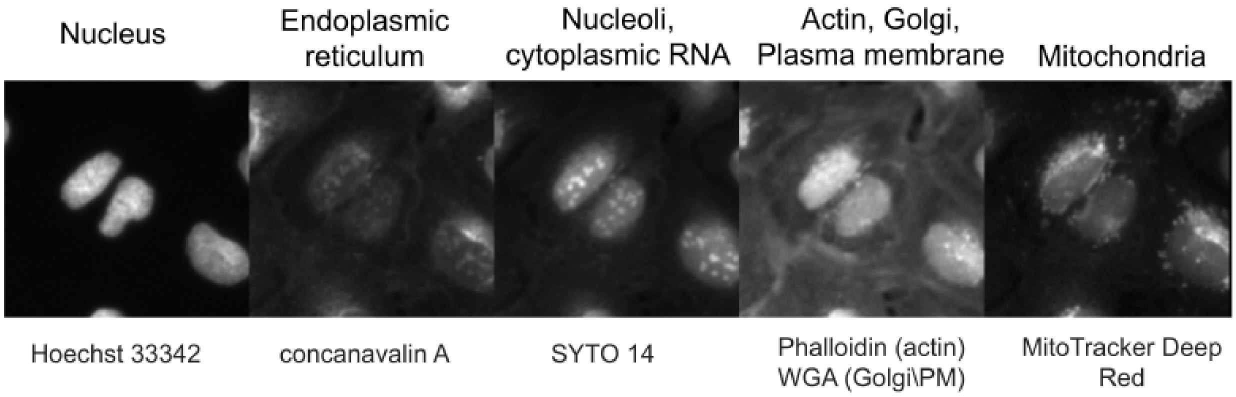


Figure 5: Example of an image obtained with Cell Painting with compartments (labels on top) and stains (labels at bottom). The image is from BBBC022 dataset [47].

CellProfiler [19] is open-source software for the quantitative analysis of cell phenotypes. It is designed for biologist-analysts, so it does not require particular experience in the field of computer science, the biologist-analyst develops only the pipeline with the modules and their settings and best practices pipelines are available for certain types of data (`https://cellprofiler.org/published-pipelines`). The output of the CellProfiler is the feature vector with human-readable features, which could be organized in the groups, such as intensity, texture and shape.

2.3 Computational methods in chemical biology

The field strongly tied to drug discovery is chemical biology, which studies the interaction of small molecules (drugs are usually small molecules) with biological systems (for instance individual cells, tissues and organisms). Like any other field, chemical biology has its own set of computational methods for different tasks [53] [54].

The first problem to solve is to represent chemical molecules conveniently and efficiently way for computational methods. There are different approaches for doing this, the one of the simplest ones is SMILES (Simplified Molecular Input Line Entry System) [55], which is simple, yet very efficient and widely used nowadays. The example is in Figure 6.

Another class of representation of molecules are the fingerprints - binary or numerical vectors of size between 16 to 16384. Fingerprints can be rule-based or obtained with deep learning methods and the efficacy of those representations is not equal [56]. The most commonly used molecular fingerprints are Morgan fingerprints [57], which are binary vectors.

Another term related to the representation of compounds is a scaffold. A scaffold is a core structure of a compound, which consists of all ring structures and links between them and was proposed by Bemis and Murcko [58], example is in Figure 6.

In the last few years, different deep learning-based approaches for computational chemistry have emerged based on convolutional or recurrent neural networks, autoencoders and graph convolutional networks [54].

One of the notable recent methods, based on graph convolutional networks is Chemprop

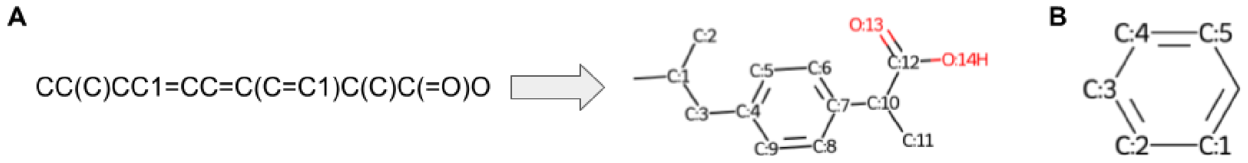


Figure 6: A. SMILES representation of Ibuprofen and its generated graphical representation. B. Bemis-Murcko scaffold of Ibuprofen. Graphical representations and the scaffold were generated with RDKit software (`https://www.rdkit.org/`).

(`http://chemprop.csail.mit.edu/`) [17] [59] [60]. It takes SMILES strings as an input (other feature vectors can be used) and reconstructs molecular graphs, where atoms are nodes and bonds are edges. Then a series of message passing steps are applied to aggregate information from neighboring atoms and bonds, to refine the representation of the molecule.

3 Summary of the research

This section contains a brief description of research projects and the results. Some details are omitted, though can be found in the related publications.

3.1 Nucleus segmentation: towards automated solutions

This section briefly discusses the content of [7].

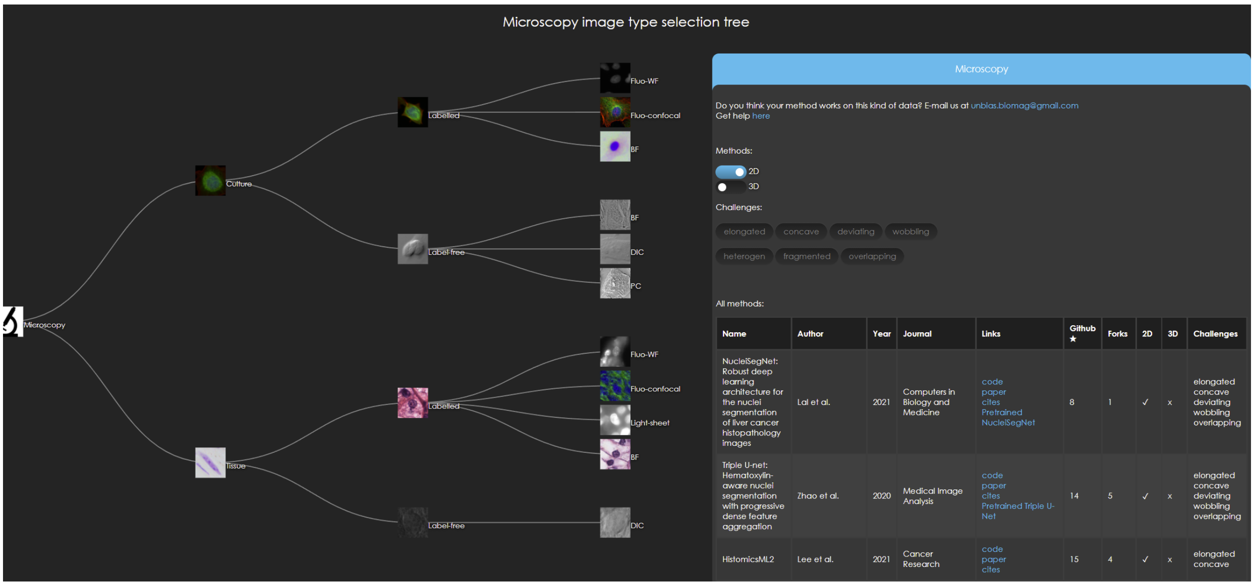
The field of nucleus segmentation was developing over the last few years with the help of deep learning. Practitioners started to use widely deep learning-based segmentation methods, especially after the DSB 2018 challenge [11], which clearly showed the superiority of deep learning-based methods over the classical ones. Besides, the computational resources have become more affordable, and the methods tend to be more user-friendly by providing guides for the tools and sometimes by providing graphical user interfaces. The review is aimed to provide an overview of the methods and datasets related to nuclei segmentation and guide practitioners in the field.

As deep learning methods require the data for training, we start the review, with the description of the openly available annotated nuclei datasets, both in 2D and 3D and for different microscopy modalities. The annotations for those datasets are shared as background-foreground (BG-FG) masks or as object masks (when each object is outlined separately). The first observation is that not so many annotated datasets are shared particularly for 3D data. The possible reason is that the laboratories started to massively switch to 3D not long ago, besides the usage of 3D over 2D is not always a necessity. An example of the 3D dataset, which might be used as a benchmark (and in fact is already used) is A549-Dataset [61]. Another observation - very few imaging modalities are well represented even in the case of 2D datasets. Most of the datasets have only fluorescent, brightfield or hematoxylin and eosin stained (H&E) images. The notable exception is the LIVECell [62] large-scale label-free dataset.

The review part about datasets is then followed by the part about annotation tools. Most of those annotation tools were released recently. We observed the presence of open-source and free tools for annotation of both 2D and 3D data.

The last part of the review is about segmentation methods and tools. The reviewed segmentation methods were classified using meaningful criteria for practitioners. First, the methods were classified by the dimensionality of the input image (2D, 3D or both 2D and 3D). Next, for each method, the availability of the code was checked. Another important criterion is the availability of extended versions of tutorials, as the users of the segmentation methods for biological tasks do not necessarily have computer science expertise and need a clear step-by-step guide to use those methods. The last important criterion is if the tool runs or can be run in the cloud, which has become a very common scenario for running computationally demanding tasks.

Another contribution of this review is the assistant tool for nuclei segmentation method selection (called *unbiased*) which is available online at GitHub Pages `https://biomag-lab.github.io/microscopy-tree/`. It is supposed to help in choosing potentially useful methods based on microscopy modality, the dimensionality of images and potential challenges in the data of interest.



Do you think your method works on this kind of data? E-mail us at unbias.biomag@gmail.com
Get help here

Methods:

● 2D
○ 3D

Challenges:

elongated concave deviating wobbling heterogen fragmented overlapping

All methods:

Name	Author	Year	Journal	Links	Github ★	Forks	2D	3D	Challenges
NucleiSegNet: Robust deep learning architecture for the nuclei segmentation of liver cancer histopathology images	Lal et al.	2021	Computers in Biology and Medicine	code paper cites Pretrained NucleiSegNet	8	1	✓	x	elongated concave deviating wobbling overlapping
Triple U-net: Hematoxylin-aware nuclei segmentation with progressive dense feature aggregation	Zhao et al.	2020	Medical Image Analysis	code paper cites Pretrained Triple U-Net	14	5	✓	x	elongated concave deviating wobbling overlapping
HistomicsML2	Lee et al.	2021	Cancer Research	code paper cites	15	4	✓	x	elongated concave

Figure 7: The interface of the assistant tool for the selection of segmentation methods. On the left, there is a tree of microscopy modalities. In the top-right, there are controls for filtering for choosing 2D/3D methods and specific methods for segmentation challenges. In the bottom-right, there is a list of segmentation methods.

The main result of the review turned out to be the raising of concerns and questions about the current state of the field. The first concern is related to the lack of diversity of existing datasets in terms of microscopy modalities. Turns out most of those openly published annotated datasets are either for H&E images or fluorescent images. Other microscopy modalities (e.g, DIC (differential interference contrast), light-sheet or phase contrast) are poorly represented in publicly available datasets. Besides, the size of the published datasets also matters, most of the datasets do not contain many objects and images.

Another point is a call for a solution to the common challenges in nuclei segmentation, such as touching, overlapping and irregularly shaped nuclei [35] [63] [64] [65]. Current deep learning methods can partially address those challenges, but more progress is desired. Both novel model architectures and high-scale training datasets might positively impact in this regard.

The real problem, which is on the surface, but rarely discussed, is the lack of a unified approach for the evaluation of nuclei segmentation methods. After inspecting all the methods eventually presented in the review, it has become clear that the evaluation methods and the datasets don't overlap. Even though there are datasets that are supposed to be the standards, different subsets of the test sets are getting used in different articles. The problem could be solved by discussions inside the community and enforcing the standards. Two candidate platforms to host such standardized tests could be Kaggle and BIAFLOWS [66].

The last conclusion of the paper is that the field could try to move towards the general models which can segment nuclei from images of diverse modalities. Some models are already capable of doing this, though with a limited amount of modalities, for instance, the models obtained during the DSB 2018 challenge [11] [67].

3.2 AnnotatorJ: an ImageJ plugin to ease hand annotation of cellular compartments

This section briefly discusses the content of [12].

To train a single-cell (nuclei) segmentation based on deep learning, annotated data is needed. To train more robust models, bigger datasets are desired, but manual annotation is an expensive process as it requires a significant amount of time and effort from biology experts. To make the annotation process faster and more accurate, a plugin AnnotatorJ [12] for ImageJ/FIJI [13] (the software for bioimage analysis) was developed which combines single-cell identification with deep learning and manual annotation.

The main feature of AnnotatorJ is a contour assistant. Contour assistant uses the pre-trained U-Net model to predict the area covered by the object of interest. After that, the user can refine the contours of the object if needed.

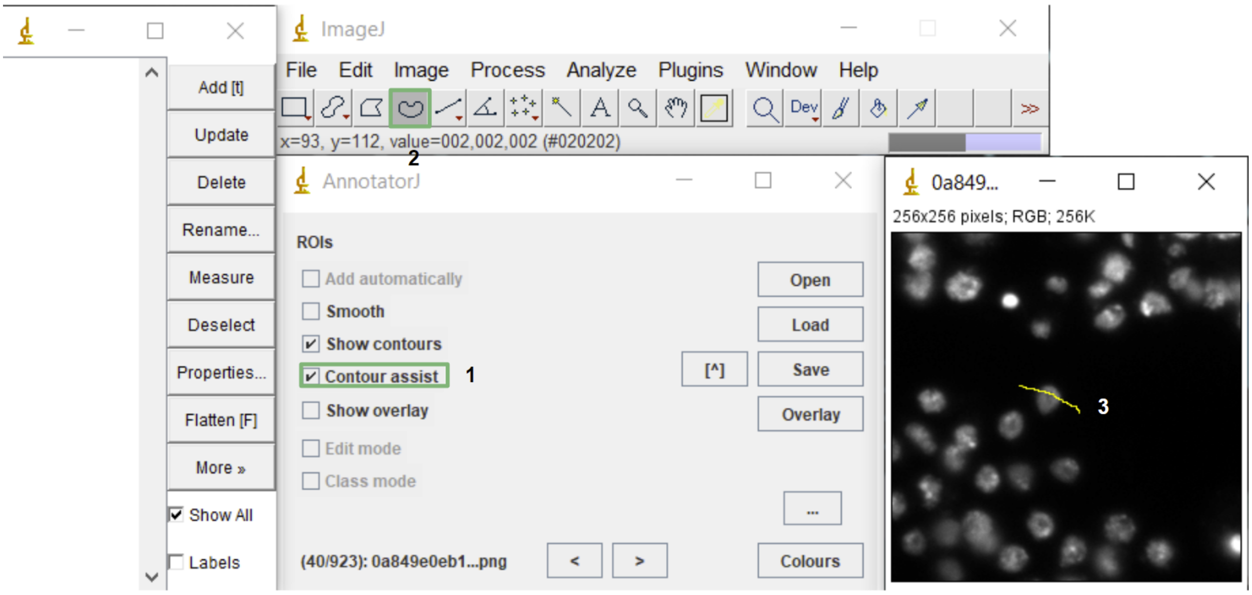
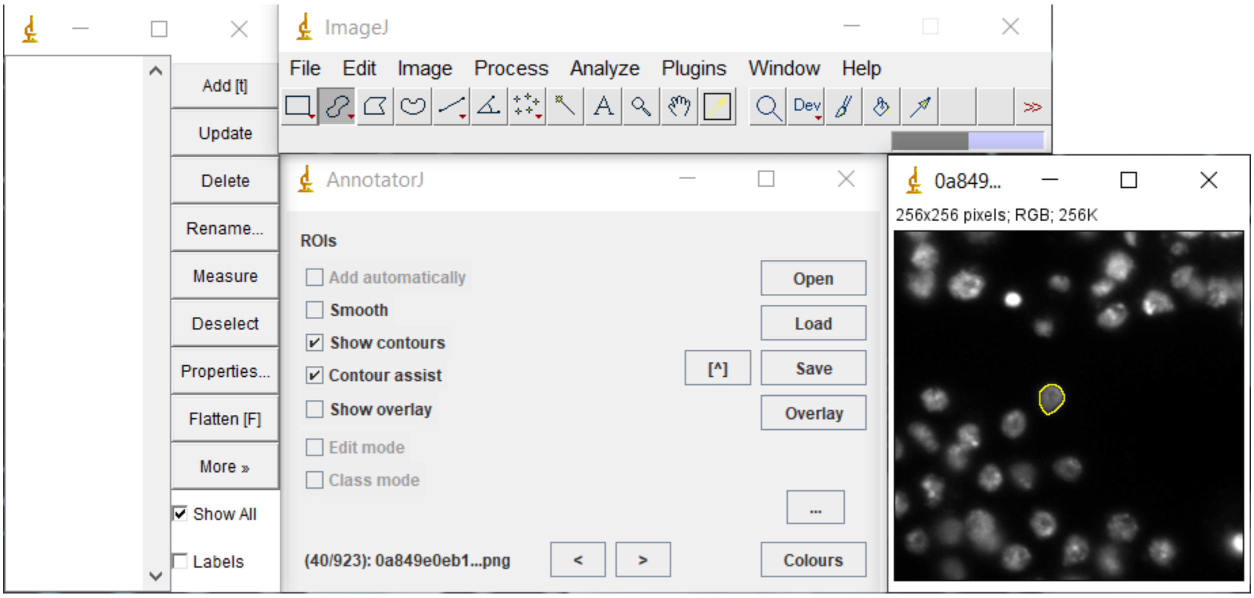


Figure 8: First step of annotation with contour assist: initialize contour by drawing a line on the object. The numbers and green boxes show the steps to perform in the interface. The source of the microscopy image: Data Science Bowl 2018 dataset [11].



☑ Show All
☐ Labels

ROIs

☐ Add automatically
☐ Smooth
☑ Show contours
☑ Contour assist
☐ Show overlay
☐ Edit mode
☐ Class mode

(40/923): 0a849e0eb1...png

Figure 9: Initialized contour by pre-trained deep learning segmentation model (in the right). The source of the microscopy image: Data Science Bowl 2018 dataset [11].

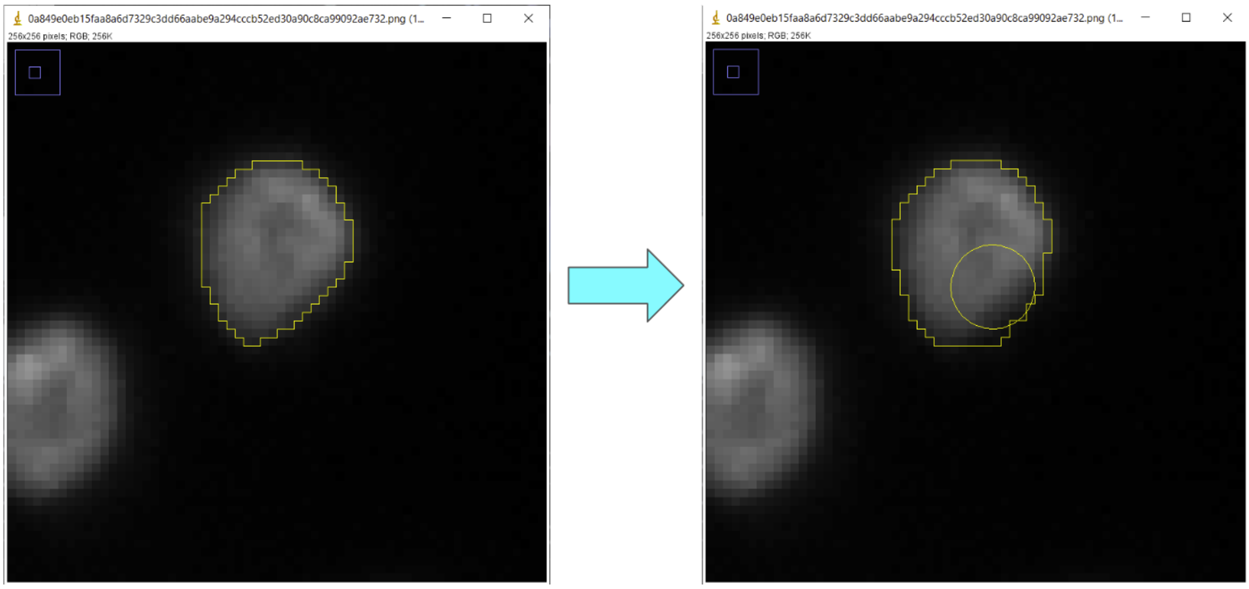
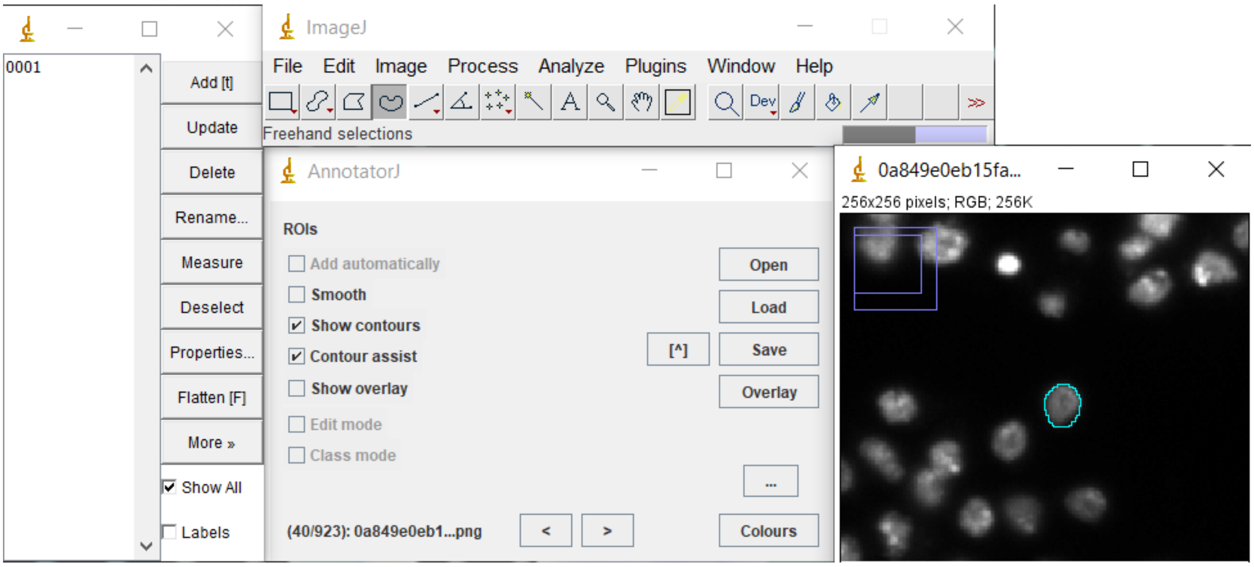

0a849e0eb15faa8a6d7329c3dd66aabe9a294cccb52ed30a90c8ca99092ae732.png (1...
256x256 pixels; RGB; 256K
0a849e0eb15faa8a6d7329c3dd66aabe9a294cccb52ed30a90c8ca99092ae732.png (1...
256x256 pixels; RGB; 256K

Figure 10: Refining the contour of the object. The source of the microscopy image: Data Science Bowl 2018 dataset [11].



☐ Add automatically
☐ Smooth
☑ Show contours
☑ Contour assist
☐ Show overlay
☐ Edit mode
☐ Class mode

Figure 11: Refined object is added as a region of interest after refining the borders and pressing 'Q' key. The source of the microscopy image: Data Science Bowl 2018 dataset [11].

To make trained models compatible with ImageJ/Fiji, which is developed in Java, we used the library DL4J and ND4J (`http://deeplearning4j.org/`). AnnotatorJ is openly available at `https://github.com/spreka/annotatorj`.

3.3 Test-time augmentation for deep learning-based cell segmentation on microscopy images

This section briefly discusses the content of [20].

Deep learning-based nuclei segmentation heavily relies on manually annotated data, which in most cases is annotated by domain experts. To increase the amount of training data and train more robust models, data augmentation [38] (see 2.1) has become a common technique in deep learning. Data augmentation is frequently used in the case of diverse or limited datasets, which is often the case in the field of nuclei and cell segmentation.

While the usual data augmentation approach is performed during the training time, the idea of another approach, test-time augmentation (TTA) (Figure 12) is to perform predictions on the original and the augmented versions of the data samples and then merge the predictions. This technique existed for some time and was successfully used in image-analysis tasks [68] [69] [70]. The experiments with test-time augmentation were conducted in the setting of the nucleus segmentation task.

3.3.1 Test-time augmentation

The pipeline of test-time augmentation includes four steps:

1. Augmentation of the original image.

2. Inference of original and augmented versions of the image.

3. Dis-augmentation: if the original image was flipped or rotated, the transformation should be reverted to the original orientation to allow further correct merging of the predictions.

4. Final merging: this step is different for Mask R-CNN and U-Net and discussed further.

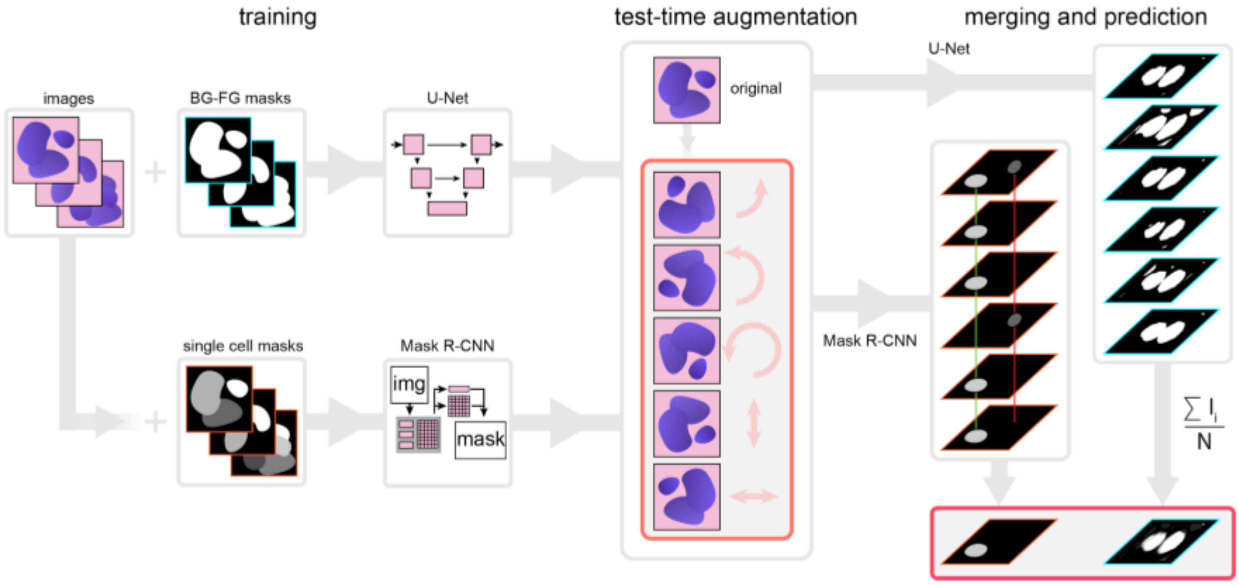


Figure 12: Proposed test-time augmentation techniques. Input: Run inference on several augmented instances of the same test images with trained models. To merge predictions, pixel-wise majority voting was used for U-Net and object matching with majority voting was used for Mask R-CNN. The source of the figure [20].

For U-Net predictions step (4) is straightforward, just sum and average all the dis-augmented probability maps. The resulting probability map is then converted to a binary mask by thresholding (0.5) which is further used for evaluation of the segmentation (Figure 12, right).

Mask R-CNN, as an instance segmentation framework, requires more post-processing. Here, each object is processed separately: for each detected object the majority voting is done. Before majority voting the object alignment should be done: the objects from the predictions of original and augmented versions of the input image are checked if those can be considered the same object. In this setup, two objects (each from different versions of the input image) are considered to be the same object if the intersection over union (IoU, also known as Jaccard Index, (Eq. 1)) between them is at least 0.5. If the same detected object is present in the majority of the predictions, then it will be included in the final prediction mask. The mask of the included object is corrected by majority voting on the pixel level.

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

3.3.2 Materials and methods

For the experiments the popular neural networks for segmentation were chosen: U-Net [33] (for semantic segmentation) and Mask R-CNN[34] (for instance segmentation) and the data for experiments mostly comes from Data Science Bowl 2018 dataset [11] with additional sources [71] [72] [73] [74] [75] [76] [77]. The original images were cropped to the size of 512 × 512 pixels. Images with a resolution lower than 512 × 512 were resized. This primary dataset was split into two datasets: one with fluorescent images (further referred to as Fluorescent or Fluo) and tissue images (further referred to as Tissue). For both of those datasets the following train-test splits were done:

- 95% images in the train set and 5% in the test set (referred to as Fluo_5 or Tissue_5)
- 85% images in the train set and 15% in the test set - repeated 6 times in cross validation setting (cross-validation split 1 is referred to as Fluo_15 or Tissue_15)
- 70% images in the train set and 30% in the test set (referred to as Fluo_30 or Tissue_30)

Separate models were trained for each holdout set. For training, the augmentation was used using horizontal and vertical flip, 90°, 180° and 270° rotations. Augmentations were done before the training (not on-the-fly), which means that the training set size was equal for each split was $6 * number\ of\ unique\ images\ in\ the\ training\ set.$

In the experiments with the U-Net (the architecture was described in 2.2.1) the widely-used implementation [78] based on Tensorflow [79] and Keras was used. The models were trained for 200 epochs with a constant learning rate of 3×10^{-4}. The initial parameters were initialized randomly. A binary cross-entropy loss function with ADAM [80] optimizer were used. Batch size was set to 1 due to GPU memory limitations. Additionally, trainings with and without the use of augmentations in training time were run with U-Net. For the experiments with Mask R-CNN, Matterport's codebase was used [81], also based on Tensorflow and Keras. Evaluation scripts were used from [37].

Mask R-CNN models were trained for 3 epochs for different layer groups in the following order:

- Initialize with COCO weights (`https://github.com/matterport/Mask_RCNN/releases/download/v1.0/mask_rcnn_coco.h5`)
- Epoch 1: all network layers were trained at a learning rate of 10^{-3}.
- Epoch 2: training of ResNet stage 5 and head layers at a learning rate of 5×10^{-4}.
- Epoch 3: Train only head layers at a learning rate of 10^{-4}.

The loss function was binary cross-entropy with ADAM [80] optimizer, batch size 1. This training strategy replicates the one from [37]. Mask R-CNN models were trained only with the use of augmentations in training.

mAP_{DSB} for an image is calculated as follows: calculate the average precision over all test images at IoU threshold t (IoU is calculated between predicted and ground-truth objects) and average over all IoU thresholds T (2). In this equation, $TP(t)$, $FP(t)$ and $FN(t)$ stand for a number of true positive, false positive and false negative objects, respectively:

$$\begin{gathered} mAP_{DSB} = \frac{1}{|T|} \sum_{t \in T} \frac{TP(t)}{TP(t) + FP(t) + FN(t)}, \\ T = \{0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95\} \end{gathered} \tag{2}$$

U-Net predictions were evaluated using the intersection over union metric (Jaccard Index) (Eq. 1). TTA's performance is evaluated by calculating the difference between the prediction scores obtained after applying TTA (*merged*) and after regular prediction (*original*). Next, TTA's performance was evaluated by calculating the difference:

$$delta = merged - original \tag{3}$$

3.3.3 Results

Test-time augmentation improved the performance for all the train-test splits on average, if used together with Mask R-CNN models. The mean gain in the mAP_{DSB} metric is between 0.01 and 0.02. While in most of the test images mAP_{DSB} improved, there are a few images with degraded performance (Figure 13).

Test-time augmentation used together with U-Net models also provided improvement in the IoU metric. We can observe that for most model checkpoints in every training scenario, except at the beginning of the training, when the model is underfitting (Figure 14).

In some test examples, test-time augmentation could change the prediction quality by a large margin (see examples in Figure 15).

Test-time augmentation combined with the method [37] (the best performing method for the DSB 2018 test set according to the Kaggle scoreboard at the time of publishing of the paper [20]) further increases the performance by 0.011 in mAP_{DSB}.

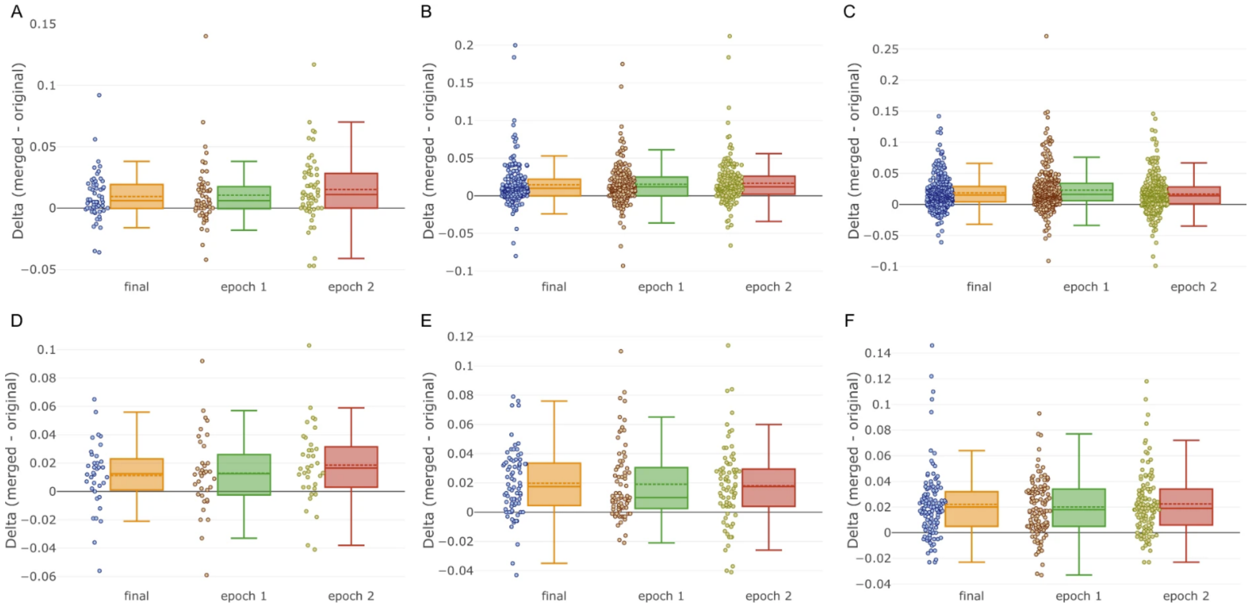


Figure 13: Test-time augmentation impact on segmentation performance (*delta* of mAP). Each point represents an image. Bars: training epochs. A dashed line in bars: mean, a solid line in bars: median. Sets: A. Fluorescent_5. B. Fluorescent_15 (cross-validation 1) C. Fluorescent_30. D. Tissue_5. E. Tissue_15 (cross-validation 1) F. Tissue_30. The source of the figure [20].

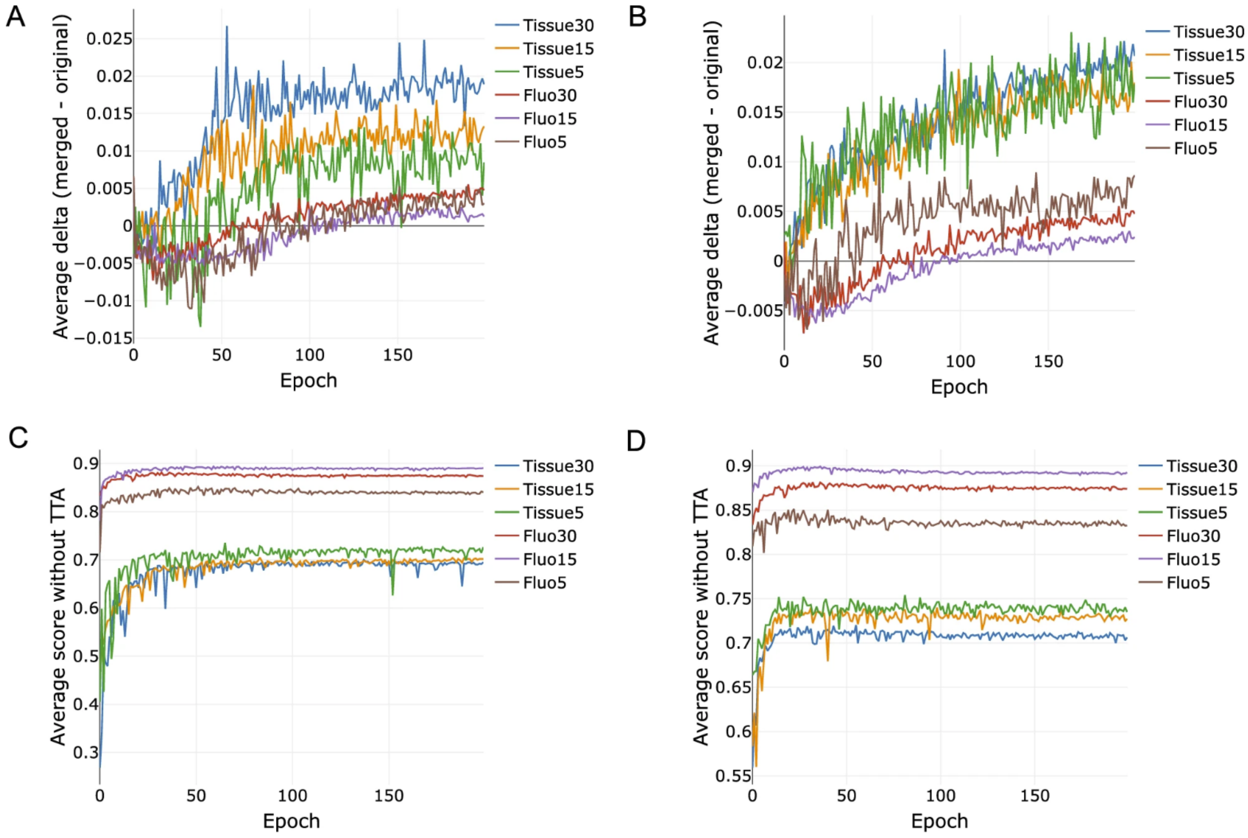


Figure 14: Mean Jaccard index in the test sets and impact of TTA for U-Net. A. Mean *delta* of Jaccard index in models trained without augmentations. B. Mean *delta* of Jaccard index in models trained with augmentations. C. Mean Jaccard index in test sets in models trained without augmentations. D. Mean Jaccard index in test sets in models trained with augmentations. The source of the figure [20].

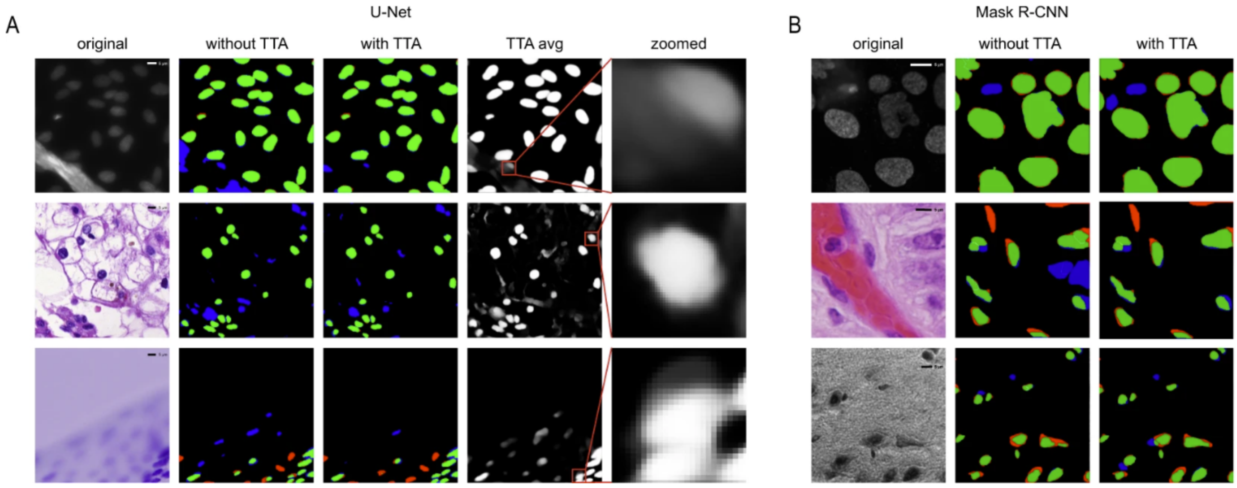


Figure 15: Comparison of predictions with and without TTA on example images. A. U-Net. First column: original image, the second: predictions without TTA, the third: predictions with TTA. Colors: false negative predictions (red), true positive (green), and false positives (blue). The fourth column – averaged TTA predictions before thresholding and the fifth: zoomed insets from the previous column. Rows are example images. B. Mask R-CNN. Columns are as first three in A, rows are example images. The source of the figure [20].

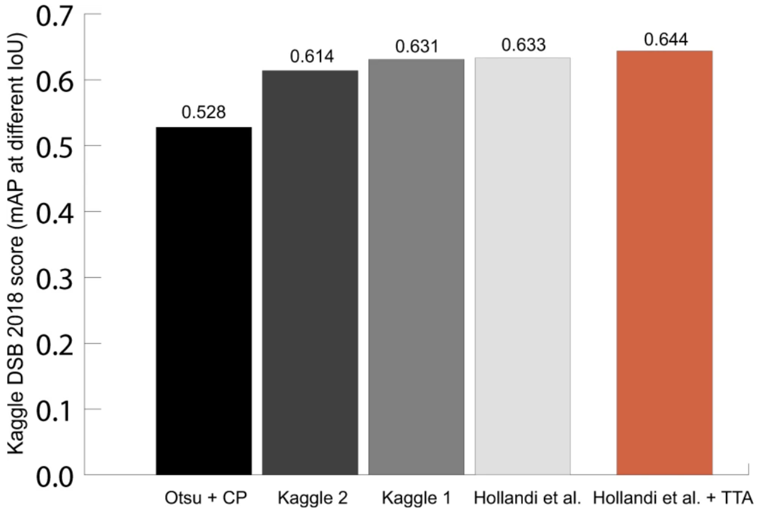


Figure 16: DSB 2018 Stage 2 test scores for different methods, compared to [37] + TTA. The source of the figure [20].

3.4 Learning representations for image-based profiling of perturbations[1]

This section briefly discusses the content of [82].

Phenotypic drug discovery is based on observations of drug effects on treated subjects, in our particular case, we consider single-cells. This problem not only requires significant

[1] The article is online as a pre-print and is being submitted to a journal

wet lab efforts but also computational approaches to process the output data. One of the first attempts to measure treatment effects using features extracted from fluorescent imaging data was [9]. Later on, CellProfiler [19], the standard approach to extract representations of single-cells was released. It produces features which are human-readable and their usefulness was proven in different downstream tasks [16].

Now, the question is, what if we can extract even more biologically relevant representations of cells from images using deep learning? With inspiration from representation learning and popular deep learning architectures for image classification, researchers have started to seek a methodology that could allow them to extract such biologically relevant representations.

One of the first attempts of using transfer learning (usage of pre-trained image classification networks with ImageNet dataset [15]) for morphological profiling was performed in [51] on full images, meaning that the full image was resized to the input size of the network and was run in inference mode.

Training models directly on images of single-cells have been explored in proof-of-concept experiments [50]. It was based on weakly-supervised learning (WSL), which does not require manually annotated data to learn feature representations. Instead, it uses treatment labels as a proxy for the phenotypes of interest. These treatment labels are weak because there is no certainty that all the treatments have a phenotype sufficiently different from the untreated cells (negative controls) or resulting phenotypes are not similar for different treatments.

Here, a systematic evaluation of three large-scale Cell Painting public datasets is conducted. Those datasets contain thousands of perturbations, hundreds of plates, and millions of single cells. The tested representations are extracted by pre-trained models and models trained in a weakly-supervised setting and compared against classical features. To run training and feature extraction experiments, the publicly available tool *DeepProfiler* was developed.

The current best practices found for making deep learning methods improve the quality of downstream analysis, which are reported below. For interpretation of the obtained results with trained models and reasoning about challenges, a causal modeling framework is used [83] [84].

3.4.1 Cell Painting datasets

In this study, five datasets were used in total:

- BBBC037 (also known as TA-ORF) dataset [85], published in 2017 to test morphological profiling using overexpression in human cells as a general approach to annotate gene and allele function.

- BBBC022 dataset [47], published in 2013, screened 1600 bio-active compounds.

- BBBC036 dataset (also known as CDRP-Bioactivies) [86], published in 2017, screened 2000 bio-active compounds.

- BBBC043 dataset (also known as LUAD) [87], testing lung adenocarcinoma variants (375 in total).

- LINCS dataset [88] screened 1300 compounds.

Images in all datasets above were taken with 20X magnification and five-channel (all captured with Cell Painting assay). The first three datasets in the list above are used as benchmarks, the latter two are only used to construct a combined Cell Painting dataset (discussed in section 3.4.5).

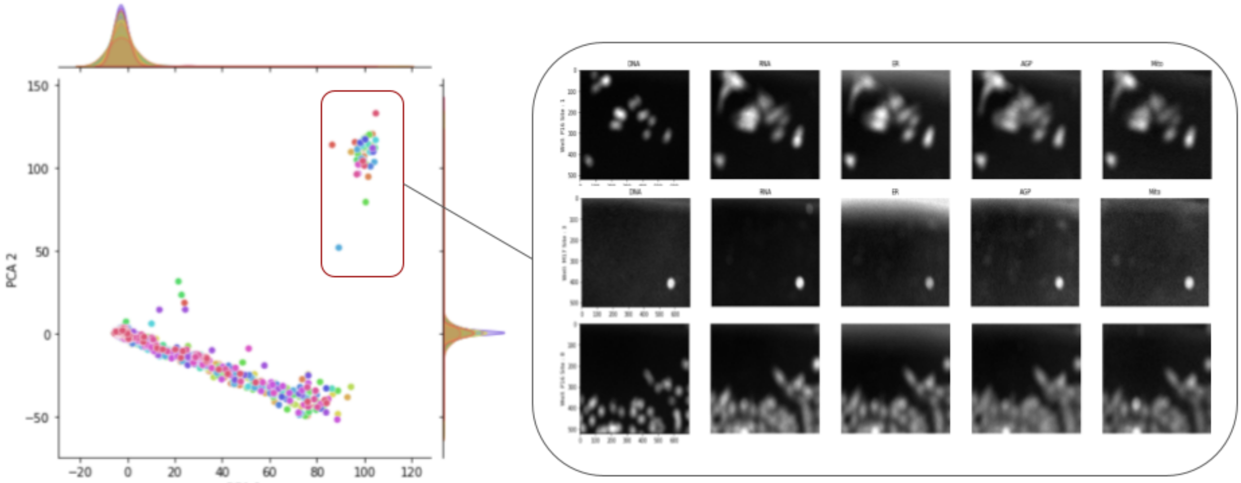


Figure 17: Example of quality controls for BBBC022 dataset. On the left: PCA plot for the first two PCs. Each point is a well, colors stand for plates. The outlier cluster is observed. On the right: examples of images from outlier wells. We see that those images are out of focus.

For the datasets above, the quality control was done to remove very noisy or out-of-focus images, as those are not able to preserve reliable phenotypic information and at the end of the day may distort the aggregated features. To do so, the features were extracted with DeepProfiler (EfficientNet-B0 model pre-trained with ImageNet dataset, see 3.4.3), then those features were aggregated by site, by well (as described in [48]) and sphering transformation (see 3.4.4.2) was applied. Then PCA was used on those aggregated profiles. The outliers observed in the PCA plots were checked manually for technical problems (Figure 17).

The datasets have class labels, such as treatments (gene perturbations in BBBC037 and compounds with concentrations for BBBC022 and BBBC036). In the case of BBBC036 and BBBC022 datasets, the treatments which were present more than once were filtered, leaving only entries with maximum concentrations. In the downstream analysis, the superclass annotations matter: gene signaling pathways (BBBC037 dataset) or mechanisms of action (for BBBC036 and BBBC022 datasets). The superclass annotations were used from [85] and then refined. Cell locations were obtained with CellProfiler.

3.4.2 DeepProfiler

A pipeline called DeepProfiler was developed which helps to train weakly-supervised models and extract representations of single-cells from high-throughput imaging experiments. DeepProfiler introduces a standardized workflow for utilizing convolutional neural networks for extracting single-cell features from large-scale image collections.

With DeepProfiler it is possible either to train the network and then perform feature extraction or to use a pre-trained network for feature extraction. The inputs of DeepProfiler are the images, corresponding metadata and the experiment configuration. DeepProfiler extracts the single-cell images (simple crops, DeepProfiler does not do segmentation on its own, but can cut objects if the segmentation mask is provided) from the full-sized images of the predefined size, and those are the inputs of deep learning networks. The workflow is shown in Figure 18. The extracted features can be used in the downstream analysis which is usually unique for a dataset and depends on the biological questions. Besides training and feature extraction, DeepProfiler has additional features for image compression and extraction of single-cell crops from full-sized images into separate image sets.

The framework is implemented in Tensorflow [79] (for both versions 1 and 2). The source code, documentation and discussions are available on the GitHub page (`https://github.com/cytomining/DeepProfiler/`).

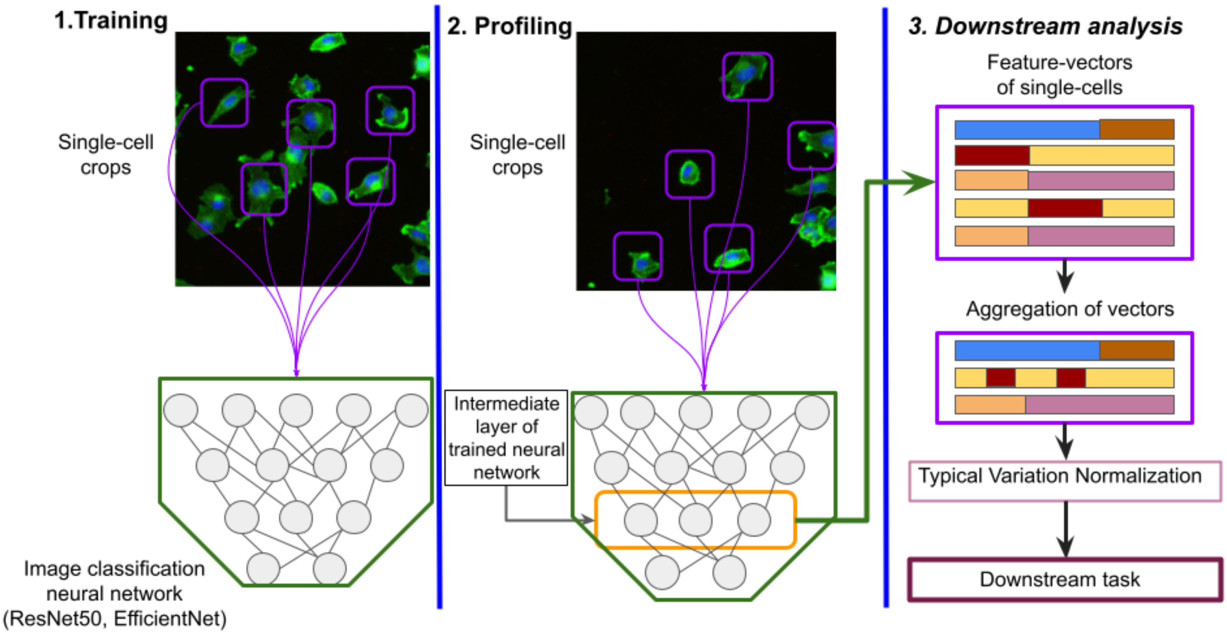

1.Training
Single-cell crops
Image classification neural network (ResNet50, EfficientNet)
2. Profiling
Single-cell crops
Intermediate layer of trained neural network
3. Downstream analysis
Feature-vectors of single-cells
Aggregation of vectors
Typical Variation Normalization
Downstream task

Figure 18: Typical usage of DeepProfiler. 1. Perform training of image classification network 2. Use the trained model to extract the representations 3. Use the representations for downstream analysis tasks. Steps 1 and 2 in the image are performed with DeepProfiler, step 3 is a user preference. The microscopy images used from the BBBC021 dataset [72].

3.4.3 Experimental setup

3.4.3.1 EfficientNet

For deep learning experiments EfficientNet [89] architecture was used, in particular the base one EfficientNet-B0. The choice was motivated by its computational efficacy and demonstrated accuracy on the ImageNet dataset [15] superior to ResNet50 [41]. EfficientNet was used in several prior publications related to cell imaging, for feature extraction and image-based profiling [87], and for training a model on a combined dataset of cellular images [90]. Some of the solutions to Recursion Pharmaceuticals cellular image classification challenge `https://www.kaggle.com/competitions/recursion-cellular-image-classification/` were based on different modifications of EfficientNet.

3.4.3.2 Experiments with pre-trained models

In this approach, pre-trained on ImageNet dataset [15]. As pre-trained networks require 3-channel input, each of the channels is replicated three times and sent to the model separately. As an input, single-cell crops of size 128×128 were used. The preprocessing for the used model also required a resize to 224×224 and min-max normalization adjusted to have a final input in the range $[-1, 1]$. The features were extracted from the $block6a\_activation$ layer. For each channel, the output dimensionality is 672 features, thus the full feature vector for the cell is 3360 features.

3.4.3.3 Experiments with weakly supervised learning

Training and the following feature extraction were conducted with DeepProfiler. The inputs are pre-cropped images of single-cells, saved as a stripe of five channels and reshaped during training, so the input to the network is $128 \times 128 \times 5$. During training the augmentations were used:

- Random crop and resize with 50% probability, the size of the crop is not less than 80% of the original size and then it is resized back.
- Random horizontal flip and then random rotation (90 degree-based).
- Color changes: brightness (up to 10% deviation from the original) and then contrast (up to 20% deviation from the original). Each channel is processed separately in both steps.

As the number of single cells varies from treatment to treatment, auto-balancing is done in each epoch of training. For all datasets, the parameters were: categorical cross-entropy loss, batch-size 32, a constant learning rate of 0.005 with SGD optimizer, augmentations on, no label smoothing and 30 epochs. The models are initialized with ImageNet pre-trained weights.

Two setups for splitting the data to training and validation were used:

- Leave-plates-out - the single-cells from one subset of plates are used for training, and from another for validation.

- Leave-cells-out - the single-cells from each plate and each well are used both in training and validation, approximately 60% of cells from each well are used in training, 40% in validation.

Using trained models, features were extracted from *block6a_activation* layer (feature vector size is 672).

3.4.3.4 Computational efficacy

The computational efficacy was estimated in terms of computation clock-time and storage space needed versus classical features. The proposed approach is faster, than the classical as it utilizes GPU parallelization. NVIDIA V100 was used for all deep learning experiments. Training time on average across the datasets takes 3.3 hours, profiling approximately takes 0.58 hours with the pre-trained model and 0.22 hours per plate with trained models. The pre-trained model takes more time as five inference passes are needed for an image. Comparison is available in Figure 19. The price is not compared here, though commonly cloud GPU computation is more expensive than on CPUs.

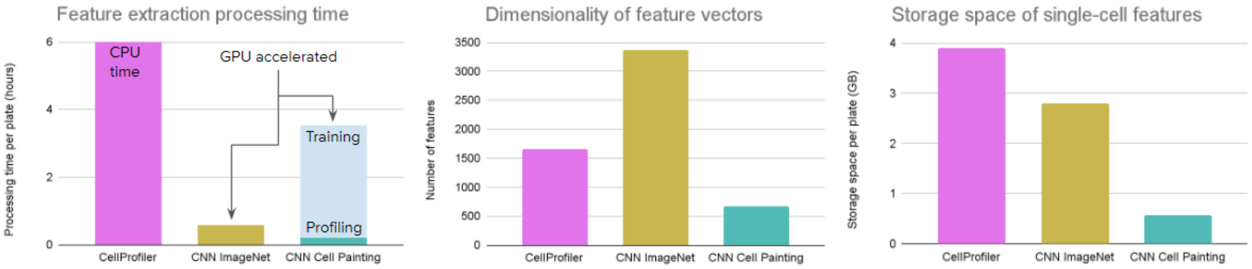


Figure 19: Computational cost of profiling strategies. The source of the figure [82].

3.4.4 Profiling workflow and evaluation

3.4.4.1 Feature aggregation and similarity matching

The feature aggregation is a pipeline to get treatment-level profiles from single-cell profiles [48]. There are intermediate levels, such as field-of-view (image)-level and well-level. The feature vectors of single-cells are aggregated using the median to image-level, and then, image-level profiles are aggregated using mean to well-level profiles. In this work, feature aggregation steps are the same, disregarding the source of the features either CellProfiler or deep-learning models.

To assess the similarity between treatments different metrics can be used [48], here the cosine similarity is used (also used in other works [91]).

3.4.4.2 Batch correction using sphering transform

One attempt for reduction of unwanted technical variation is Typical Variation Normalization (TVN) proposed in [92], also used in [93]. It computes axes of variation using principal component analysis on negative control well-level profiles. The obtained axes are normalized, which makes axes of large variation be reduced and axes of low variation to be amplified. The normalization transformation is then applied to all well-level profiles.

Here, the ZCA-sphering transformation is used similarly as TVN. As an input, the matrix of well-level negative control features $X^{n \times d}$ is used, where n is the number of control wells and d is the feature vector dimension. The covariance matrix for X is $\Sigma = \frac{X^T X}{n}$, its eigendecomposition is $Q = U \Delta U^T$, where Δ are eigenvalues. To obtain a final ZCA-transformation [94] (sphering), is the following $U(\Delta + \lambda)^{-1} U^T$, where λ is a regularization parameter.

3.4.4.3 Evaluation and metrics

As described in [16], the evaluation task is to check if the most similar treatments according to the similarity metric belong to the same gene pathway or mechanism of action (MoA). Several metrics were used for evaluation, all of them briefly described below. In further text, *query treatments* are referred to as treatments which have at least two treatments in the same MoA or pathway and are used as queries in the ranking task.

First metric that was used is folds of enrichment. The odds ratio is calculated, similar to [16], the main difference is that here it is done only for 1% threshold and this is done for each query treatment separately. Then, the simple mean of obtained values is computed.

As another metric, an interpolated precision-recall curve and mean average precision for the ranking task was used. This metric is calculated in the following way: each treatment is a query, and the top similar treatments to the query treatment are checked. Precision@K in this ranking task is the ratio of treatments that belong to the same MoA/pathway as the query out of the top K most similar treatments. The same intuition is applicable for *Precision@Recall*: for one treatment (query) we go through all the treatments ranked by distance until we reach a recall of 1 (find all positive matches). As each MoA/pathway has a different number of associated ground truth treatments, *Precision@Recall* is interpolated to cover the max number of recall points, interpolated precision is defined as $p_{interpolated}(r) = max_{r' \geq r} p(r')$ [95]. Average precision (area under interpolated Precision-Recall curve) is a mean of $p_{interpolated}$ at all recall points. mAP here is a simple mean of average precisions for individual queries.

Hits in the top 1% metric simulates the task of finding a 'hit' in the most promising candidate treatments. The metric is applicable on several levels of profiling:

- Treatment-level: measure the number of query treatments which have a treatment (response) with the same MoA/pathway among the top 1% of most similar response treatments.

- Well-level: The well profile is used as a query (all treatments can be used). The number of treatments, which have query wells with the response wells of the same treatment among the top 1% of most similar wells is computed.
- Image-level: The image profile is used as a query (all treatments can be used). The number of treatments, which have query images with the response images of the same treatment among the top 1% of most similar images is computed. The images of the same well as query image are excluded from the possible responses.

3.4.5 Strong treatment selection and combined Cell Painting dataset

To expand the potential feature-space both with biological and technical variation the treatments resulting into a strong phenotypes were collected from five Cell Painting datasets. Strong treatment here is defined as one to produce a phenotype which is different to a phenotype of untreated cells. To estimate the strength of the phenotype, CellProfiler feature space is used (batch-corrected with regularization parameter $1e-2$) and measure the Euclidean distance between the well-level profiles of treatments and negative controls (Algorithm 1).

Algorithm 1 Strong treatments selection

```
for each $p$ in $Plates$ do
    Calculate median profile of negative controls in the plate - $MCP_p$
    Calculate Euclidean distance between the treatment well-level
  features and $MCP_p$, get the distances $EDT_p$
    Calculate Euclidean distance between the negative control well-level
  features and $MCP_p$, get the distances $ECT_p$
    Calculate $\mu$ and $\sigma$ of $ECT_p$
    Use $\mu$ and $\sigma$ to Z-score $EDT_p$
end for
for each $t$ in $Treatments$ do
    $Z(t) \leftarrow \sum_p^{Plates} EDT_p(t)$, where $Z$ stores the final distances for each
  treatment
end for
```

Selection of the strong treatments for the combined Cell Painting dataset did include the following steps:

- Select top 500 strongest treatments according with Algorithm 1 from BBBC022.
- Intersect those with BBBC036, include the intersection into the combined Cell Painting dataset.
- Additionally select 50 from BBBC022 and 62 from BBBC036 strongest treatments and add them to the dataset.

- Select 7 random treatments from LINCS, from the top 20 (by a number of associated treatments) MoAs, and add them to the dataset.
- Select 28 overlapping wildtype genes between BBBC043 and BBBC037 dataset and add to the dataset.
- Additionally select 29 strongest treatments from BBBC037 and 32 from BBBC043 and add them to the dataset.
- Filter out classes with less than 100 cells.
- Add controls one class for compound screening datasets (BBBC022, BBBC036, LINCS) and another for gene overexpression datasets (BBBC037, BBBC043). Control cells from BBBC036 and LINCS are partially selected.

Resulting dataset contains 8.3 million single cells from 232 plates, 488 treatments and 2 types of negative controls. More information about the dataset is in the Figure 20.

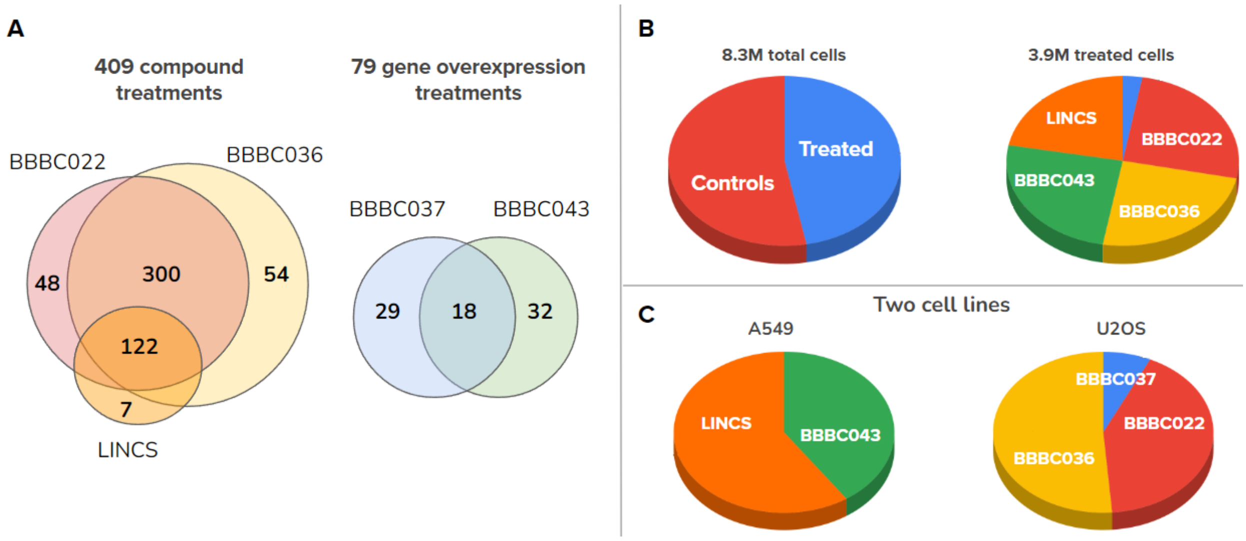


Figure 20: Description of combined Cell Painting dataset. A. Treatment sources in the combined dataset. B. Treated vs control cells distribution and sources of treated cells. C. Sources of cells inside per cell line. The source of the figure [82].

3.4.6 Causal relations in screening experiments

By applying different treatments to cells, biologists are trying to perturb their state and observe the response. The causal graph for that kind of experiment includes four variables: treatments T, images O, phenotypes Y and batch-effects C. In causality modeling terms, those are interventions, observations, outcomes and confounders respectively. T and O are observed variables, while Y and C are latent variables. The goal is to learn Y, a multidimensional representation of treatment, which could be used in the further downstream task. To be useful in the downstream analysis task, Y should encode biologically relevant representation, though the reality is that technical variation, the batch-effects C affect all other elements of this causal model. C affects images by technical variation in the image acquisition process, treatments by plate-layout design (the template of the positioning of treatments in plates in the screening experiment) and phenotypes by environmental conditions. The relations are shown in the graph (Figure 21).

Treatment is expected to be the main cause to change in the phenotype of the cell. To extract the representation of phenotypic outcome, WSL is used with the pretext task of treatment classification. The representations extracted from the intermediate layers of CNNs encode all visual variation, in this case, both batch-effects and phenotypes. WSL together with batch correction would help to disentangle phenotypic variation from technical.

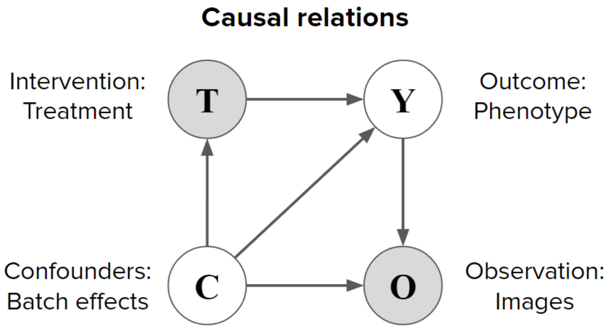


Figure 21: Causal model for screening experiment. T stands for treatments (interventions), O for images (observations), Y for phenotypes (outcomes) and C for batch-effects (confounders). The source of the figure [82].

3.4.7 Results and observations

The subsection discusses the results obtained with WSL on the combined Cell Painting dataset *CNN Cell Painting* model and models trained on the benchmark datasets. Pretrained model on ImageNet (also referred to as *CNN ImageNet*) dataset and classical features extracted with CellProfiler serve as baselines.

3.4.7.1 Learned representations sharpen biological features

CNN Cell Painting model performs better in quantitative evaluation than both baselines in the evaluation task (Figure 22, cyan points). That was expected as manually engineered features might miss some information and the ImageNet model is trained on a completely different domain and not optimized for the images of cells. The models trained only on the corresponding benchmark datasets did not show a consistent improvement in their performance against the baselines (Figure 22, green points).

For qualitative assessment, UMAP projection [96] of feature space obtained with *CNN Cell Painting* was used (Figure 23). In BBBC037 dataset, treatments are grouped together according to their pathway annotations, reproducing observations from [85]. In BBBC022 and BBBC036 projections, many treatments are also grouping together according to their MoAs.

CNN ImageNet demonstrates similar or lower performance compared to CellProfiler features (Figure 22, yellow and pink points).

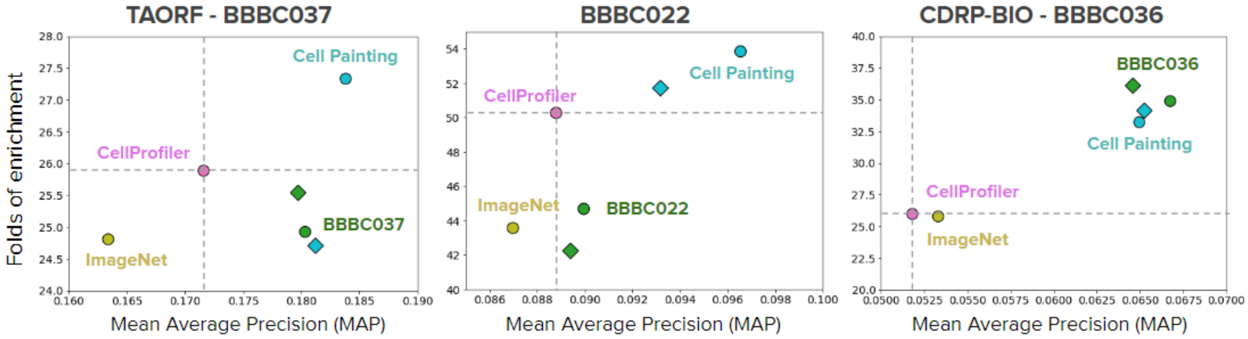


Figure 22: Quantitative performance of feature representations for three benchmark datasets in two metrics: mean average precision (X-axis) folds of enrichment (Y-axis). On the plot, the baselines are CellProfiler (pink) and CNN ImageNet (yellow), trained models: CNN Cell Painting model (cyan), trained on corresponding benchmark dataset (green). Leave-cells-out training-validation scheme shown with circles and leave-plates-out with diamonds. The source of the figure [82].

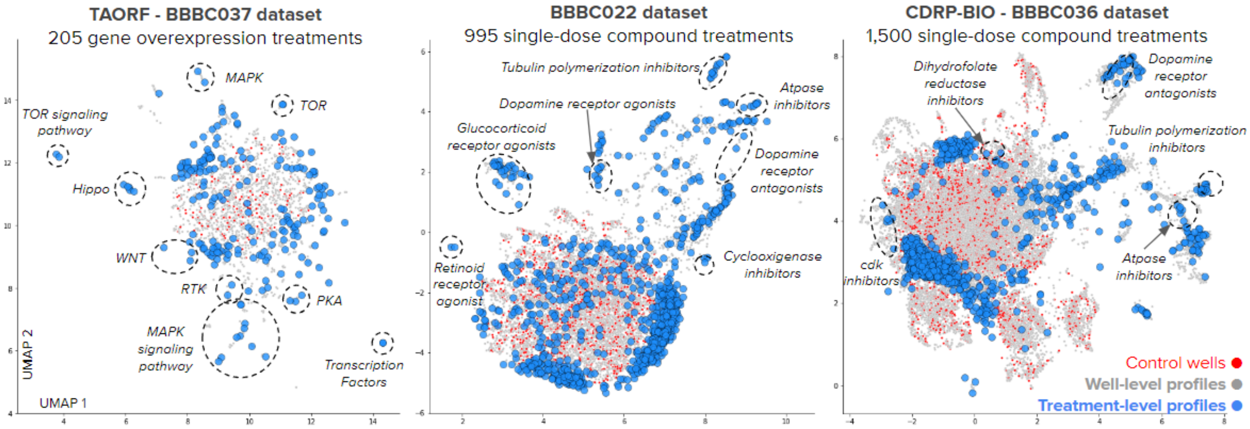


Figure 23: UMAP plots of well-level features extracted with Cell Painting CNN for three benchmark datasets. Gray points: well-level profiles of treatments, red points: well-level profiles of negative controls, blue points: treatment-level profiles. Dashed ellipses highlight clusters of treatment-level profiles with the same biological annotation. The source of the figure [82].

3.4.7.2 WSL learns both the phenotypes and the batch-effects

Different validation schemes leave-plates-out and leave-cells-out (see Experimental setup 3.4.3) help to understand the information contained in features learned from Cell Painting images. In leave-cells-out validation scheme the model as access to the full distribution of biological variation (treatments T) and technical variation (batch-effects C), yet with leave-plates-out scheme, the model still has access to the full distribution of biological variation, but only to a part of technical variation.

Major performance difference was observed in the pretext classification task for those two validation schemes. In leave-cells-out setup, the trained CNN can accurately classify single-cells from both training and validation sets, while in leave-plates-out setup, the trained model completely fails to classify single-cells in validation set (Figure 24). Nonetheless, two models trained with different validation schemes demonstrate similar performance in the downstream task (Figure 22). This observation leads to a conclusion that WSL models try to take advantage of any information that can explain the link between the images and treatments, including batch-effects. The validation performance in leave-cells-out is too optimistic (batch-effects are heavily used to build the link between observation and intervention), on the contrary, leave-plates-out validation performance is too pessimistic as in this case the model is not aware of confounding variation in validation plates.

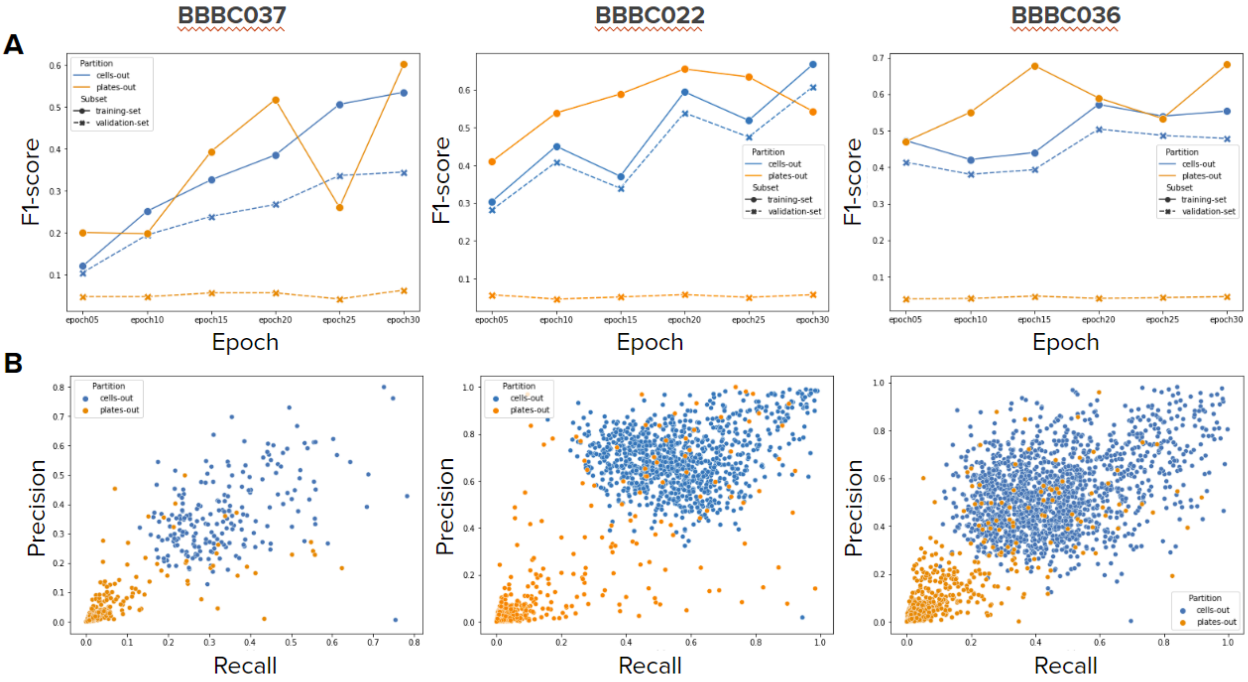


Figure 24: Classification performance in the pretext task (treatment classification) in the benchmark datasets for leave-plates-out (orange) and leave-cells-out (blue) training-validation schemes. A. F1-score for the training set (solid line) and validation set (dashed line) for every fifth epoch. B. Recall (X-axis) and precision (Y-axis) for the final checkpoint. Every point is a class (treatment, including negative control). The source of the figure [82].

3.4.7.3 Learning with strong phenotypes improves performance in the biological task

As in the previous section it was observed that controlling the distribution of confounding factors C does not change the downstream performance, now it is time to explore what happens if the phenotypic distribution Y is restricted. The intuition is that WSL minimizes an error in the pretext task by exploiting confounding factors to correctly classify treatments with a weak phenotypic response. Such treatments might have a stronger technical signal rather than a biologically relevant phenotypic signal.

The strong treatments were selected by measuring Euclidean distance between negative control and treatment profiles, obtained with CellProfiler (see section 3.4.5). That is an approximation of average treatment effect (ATE), a causal parameter for intervention outcomes. As we cannot observe the untreated (control) and treated conditions in the same cell, this can be considered only as an approximation of ATE. CellProfiler features were chosen to estimate ATE as those are non-trainable, thus can serve as independent prior.

WSL training only on strong treatments only in benchmark datasets was evaluated in leave-plates-out training-validation scheme. The results demonstrate minor performance improvement against training on full datasets (Figure 25, blue points).

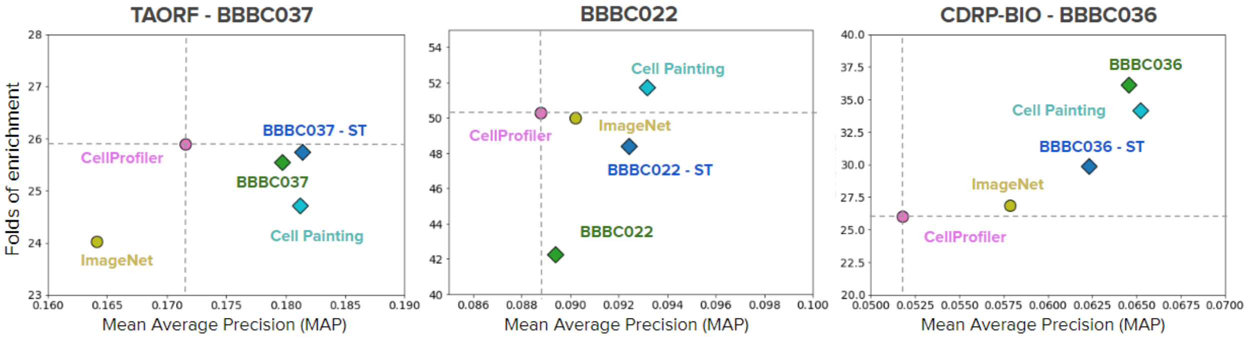


Figure 25: Quantitative performance of feature representations for three benchmark datasets in two metrics: mean average precision (X-axis) folds of enrichment (Y-axis). On the plot the baselines are CellProfiler (pink) and CNN ImageNet (yellow), trained models: CNN Cell Painting model (cyan), trained on corresponding benchmark dataset (green), trained on strong treatments from corresponding benchmark dataset (blue). All training experiments used leave-plates-out training-validation scheme. The source of the figure [82].

3.4.7.4 Diverse experimental conditions result in improved representations

The combined Cell Painting dataset was created to maximize both phenotypic (Y) and technical (C) variation by combining the treatments with the strongest resulting phenotypes from five datasets. Training on this dataset consistently improves performance over other approaches (Figure 22, cyan points), which means that this model can disentangle Y and C more efficiently. The most important outcome is that this model was trained once and could be used at all benchmarks without additional training.

3.4.7.5 Batch-correction is a crucial post-processing step

The role of batch-correction (see Batch correction using sphering transform 3.4.4.2) is to reduce the impact of confounding technical factors C. It is crucial for all representations tested: classical features, features extracted with pre-trained and trained CNNs. Mean average precision improves up to 90% versus raw features (see Figure 26). Also, using the effect of batch-correction can be observed qualitatively (Figure 27). Still, this does not mean that the batch-effects are eliminated and further research is needed to learn how to disentangle technical and biological information in representations.

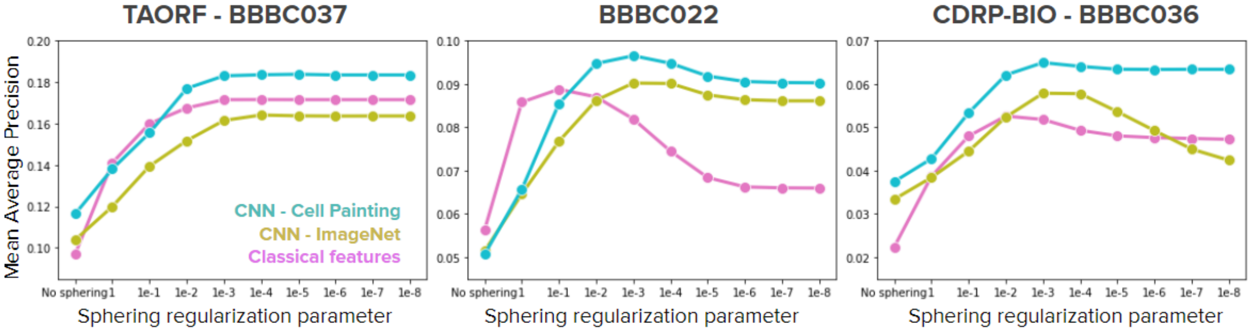


Figure 26: Mean average precision for sphering with different regularization parameters (smaller regularization term, more correction applied) for three datasets. For each dataset CellProfiler features (pink), ImageNet CNN (yellow) and Cell Painting CNN (cyan) are evaluated. The source of the figure [82].

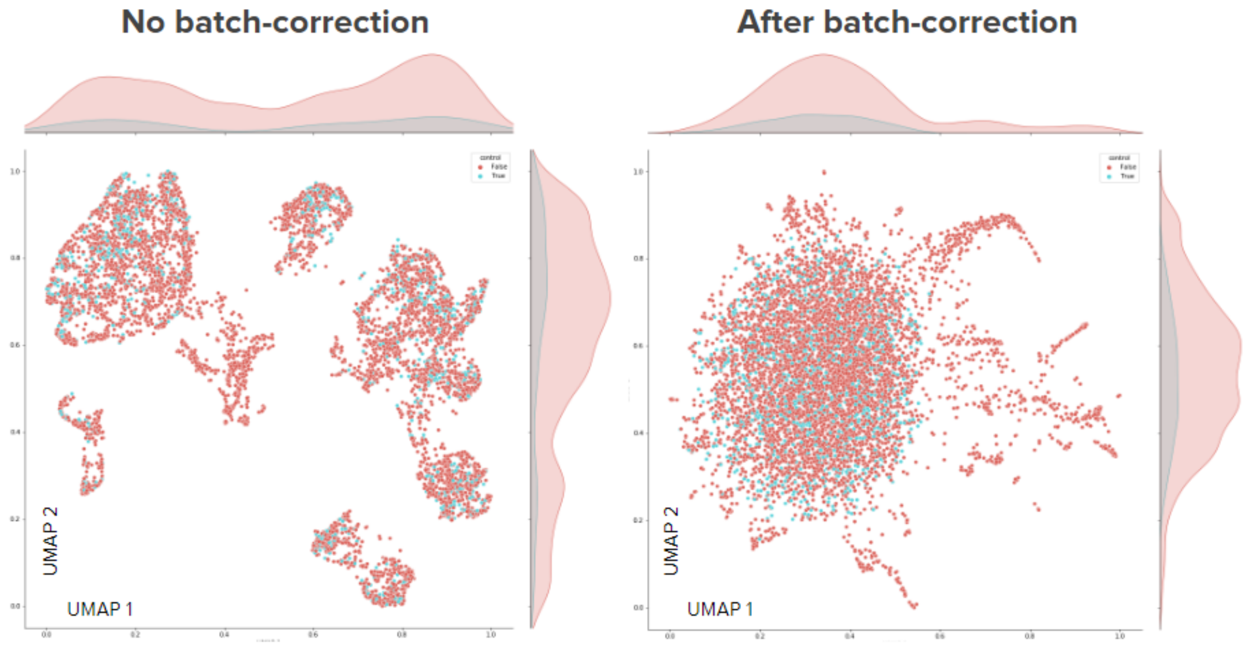


Figure 27: The qualitative effect of batch-correction in the UMAP plots. The left plot shows the UMAP representation of the BBBC022 dataset without batch-correction and the right plot after batch-correction. The points are the embeddings of well-level profiles (cyan - negative controls, red - treatments). Density plots are on the top and the right sides of the plots. Features were extracted with *Cell Painting CNN* model.

3.5 Predicting compound activity from phenotypic profiles and chemical structures[2]

This section briefly discusses the content of [97].

Drug discovery is an expensive and very slow process, there are too many theoretically possible compounds to test in a real physical experiment. Even though pharmaceutical companies may afford to test millions of compounds in their experiments, this only covers a small fraction of possible compounds. Besides, to test those compounds the expensive (as those contain valuable biological materials: primary cells, antibodies, etc.) phenotypic assay systems are used to identify candidate compounds. Finally, this process is time-consuming and requires the time of experts to run the assays.

To reduce the costs of screens in drug discovery, there is possible room for computational methods, for instance, modern deep learning might allow accurate prediction of assay activations for compounds. The previous works tried to use machine learning methods with morphology data only [98] [99].

In this project, the aim is to evaluate the predictive power of the representations of chemical structures, cell morphology profiles and gene expression profiles, to predict assay outcomes computationally at a large scale. The hypothesis is that the predictive capabilities of those data sources are complementary and those data sources could be used together to further increase the success rate of the drug screening process. Besides, the basic data fusion techniques are tested, although it is not the focus of the project and this question might be investigated further.

3.5.1 Materials and methods

The dataset is composed of four parts: assay-compound interaction matrix, morphology profiles, gene expression profiles and representations of chemical structures. All the information was collected from assays from the drug discovery experiments conducted at Broad Institute [86].

Assay-compound interaction matrix is the main piece of the dataset. Rows are compounds (represented as SMILES strings) and columns are assays. The cells are filled with 1 (hit) and 0 (no hit) and can be blank (this compound was not tested with the assay). “Hits” and “no hits” combined are also referred to as readouts. Only a fraction of compounds was tested in each particular assay, which means that the matrix is quite sparse. Initially, the matrix contained 496 assays, but filtered using the following procedure:

- Applied all pan-assay interference (PAINS) filters [100] implemented in RDKit, which removed 786 compounds, resulting in 16,210 compounds.
- Removed all assays without hits, thus the number of assays decreased from 496 to 437.

[2] The article is online as a pre-print and submitted to a journal

- Calculate intersection-over-union (IoU) for the hits between assays to find out the assays which carry the redundant information. The IoU matrix (437 × 437) was thresholded by 0.7 and then hierarchical clustering was applied with the cosine distance metric, which was used for filtering.
- Final removal of frequent hitters, defined as compounds that are positive hits in at least 10% of the assays (30 assays or more) and final cleaning of assays without any hit. In the end, the final dataset consists of 16,170 compounds and 270 assays.

Most of the assays in the final dataset are cell-based, other represented types of assays are biochemical, bacterial and yeast assays and also there are poorly represented categories of assays, such as fungal, homogeneous, viral and worm (Figure 28).

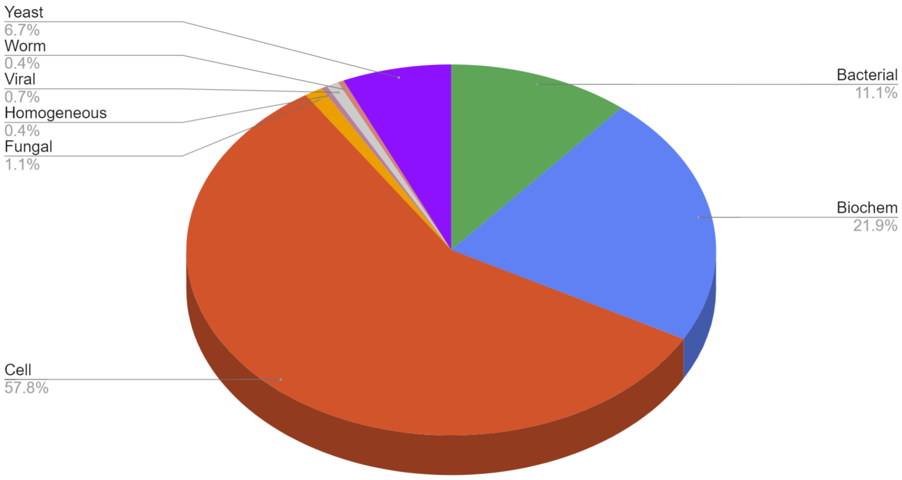


Figure 28: Distribution of the assay types in the final dataset. The source of the figure [97].

The Cell Painting assay [10] [47] [101] [102] experiments were run to obtain high-resolution five-channel images. Those images were processed with CellProfiler software to segment and obtain ∼ 1700 morphological features at the single-cell level. Those were then aggregated to the well-level as in [48]. On the well-level profiles, sphering (see also 2.2) was applied to correct for batch effects. To calculate the sphering transformation, DMSO wells from all plates were used. Then the profiles were aggregated to the treatment level (referred to as MO, except for Table 3.5.2 and Table 2). The experiments were also performed with the features without sphering, though the additional performance boost gained for the morphological features, in that case, may be biased by batch effects (Figure 29).

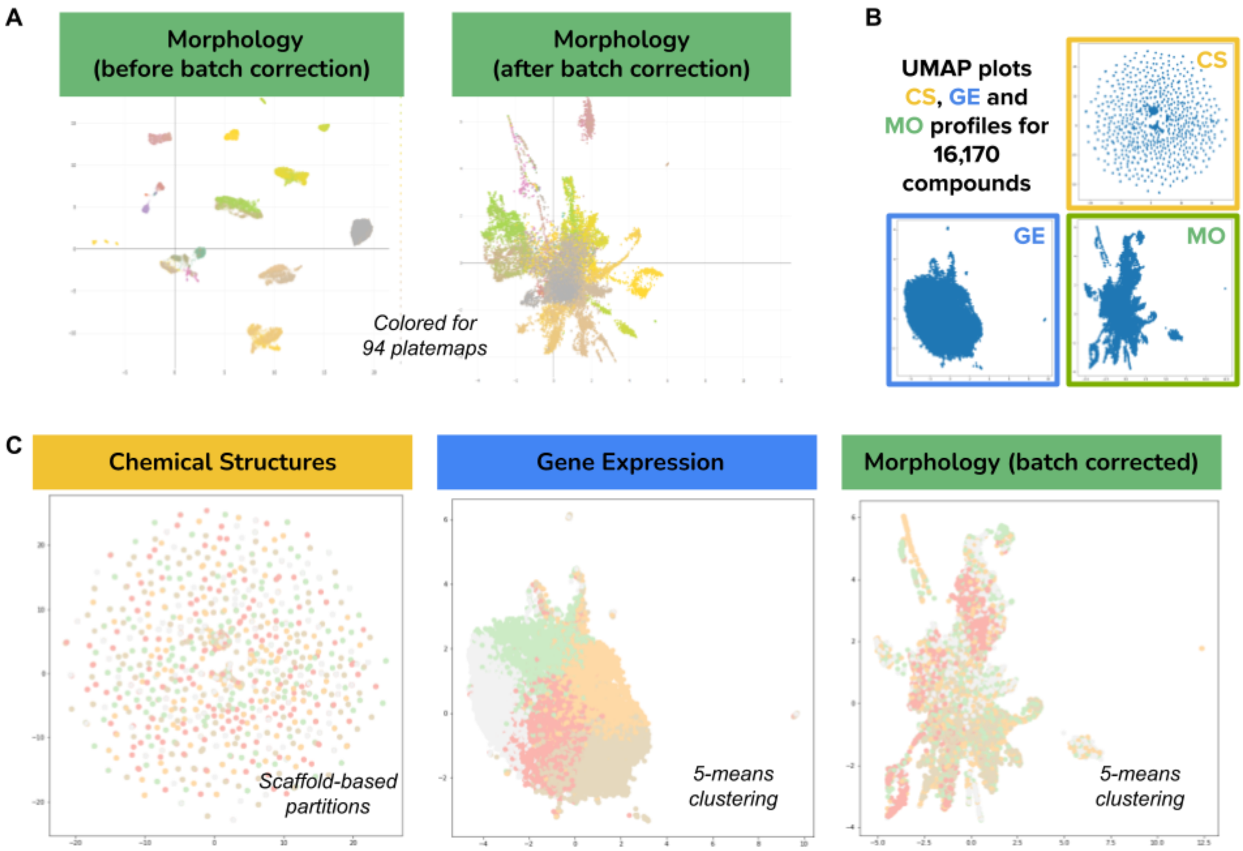


Figure 29: Compound embeddings in three different modalities. Visualizations are built with UMAP. A. The morphology feature space originally was grouped by technical variation (plate maps), which was corrected using the sphering. The color palette for the 94 plate maps is continuous and may have similar tones for consecutive plates. B. Compound embeddings in three different modalities C. The same embeddings as in B, colored by clusters obtained for cross-validation experiments (see "Experiments and results section"). The source of the figure [97].

3.5.2 Experiments and results

The experiments were conducted for several train-test split approaches. All the train test approaches share the same idea that we want to predict assays-compound interaction for compounds that are distinct relative to training data. From the practical perspective, there is little value in searching for similar chemical structures for the one with known activity. The closest train-test split to such a real-world scenario is a scaffold-based split (for 5-fold cross-validation) achieved with Bemis-Murcko clustering [58] [103].

In addition to scaffold-based train-test splits, the splits based on morphological and gene expression features are constructed. For gene expression-based splits the gene expression features were clustered and for morphology-based splits the batch-corrected morphology features were clustered (for 5-fold cross-validation) using same-size K-Means clustering (implementation [104]), see clustering in Figure 29.

As a primary metric, the area under receiver operating characteristic curve (AUROC) was used. The main results are reported for 0.9 threshold as it was used in earlier works about assay-compound activity prediction [60] [105] [106]. As a secondary metric, an area under the precision-recall curve (AUPRC) was used.

The models were trained with a logistic regression loss function for each assay, and total loss is a sum of losses for each assay. The mini-batch contains information about 50 compounds. If there is no ground-truth readout for assay-compound interaction, it is ignored for gradient update. In each training, the hyperparameter optimization was run before the training (see 3.5.1).

Our results show that morphology could accurately predict the largest number of assays with the median $AUROC > 0.9$ over cross-validation splits (28 for morphology, 19 for gene expression and 16 for chemical structures), see Figure 31. Although, for lower AUROC thresholds (0.7) chemical structures tie with morphology (also see Figure 33). Interestingly, all three modalities share zero well-predicted assays (Figure 31) and each pair of modalities share a few common well-predicted assays, which means that different data sources contain significantly complementary information.

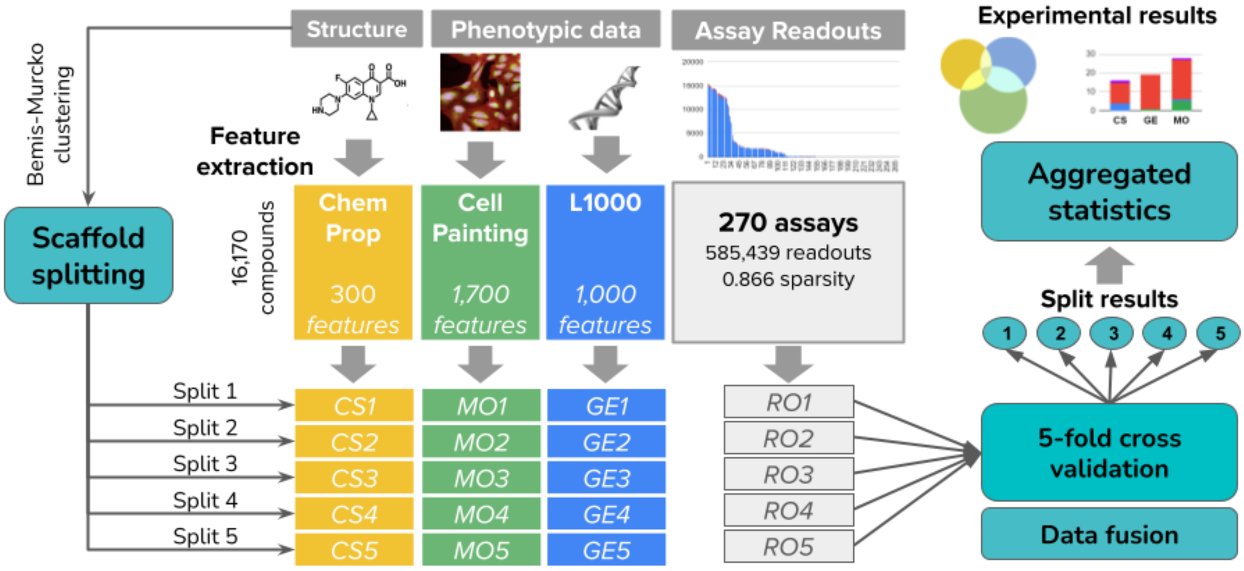


Figure 30: Illustration of experimental setup. The source of the figure [97].

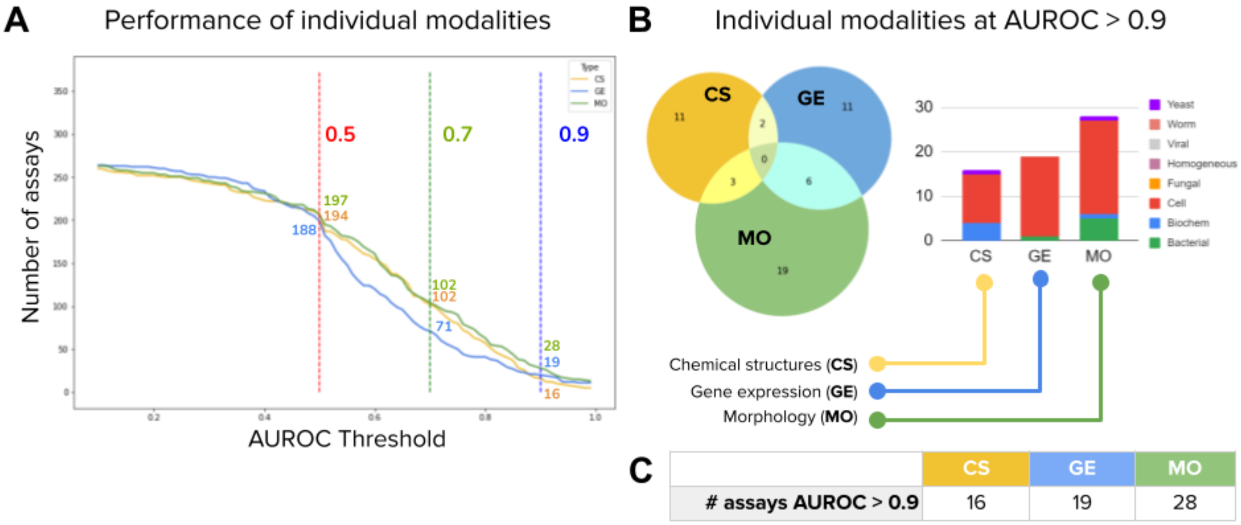


Figure 31: A. Performance of individual modalities measured as the number of assays (vertical axis) predicted with AUROC above a certain threshold (horizontal axis). B. The Venn diagrams show the number of accurate assays (median $AUROC > 0.9$ over cross-validation splits) that are common or unique to each profiling technique. The bar plot shows the distribution of assay types correctly predicted by single profiling modalities. C. Number of well predicted (median $AUROC > 0.9$ over cross-validation splits) assays by each modality. The source of the figure [97].

Not only one modality can be used for predicting the assay-compound interaction. To combine modalities into a single predictor, two approaches were used: a) ***Early fusion*** - the feature vectors are concatenated into a single vector and used as an input for the neural network. b) ***Late fusion*** - for each modality the separate model is trained and then the prediction scores are aggregated, using the maximum probability among predictions for each compound-assay pair.

According to our experiments (Table 2), early data fusion did not provide any additional performance, in fact, it did hurt the performance. Our results for individual modalities did show that they do not share many well-predicted assays in common (Figure 31), and when the feature vectors are combined, additional noise to the assays is introduced, as assays can be well predicted by one modality but cannot be predicted by another. Late fusion works better in practice, though according to the results, the performance gain is minor at best (31 well-predicted assays with CS+MO combination vs 28 with MO only). The fusion approaches in the demonstrated tests are quite simple and more investigation for more effective fusion techniques is needed. As an additional metric, retrospective performance was measured. It is a simulation of the best possible data fusion. In this analysis, know the predictions are known in advance. Usage of fused with individual modalities can give 7-17% of performance boost (Figure 32).

	Scaffold-based splits — Real world setting					
Avg. assays tested: 233.2	MO	MO-BC	GE	GE-S	CS-GC	CS-MF
Mean AUPRC	**0.261**	0.252	0.234	0.231	0.232	0.223
Mean AUROC	**0.657**	0.637	0.592	0.587	0.630	0.610
$AUC > 0.5$	**160.0**	151.4	139.2	138.8	150.2	146.8
$AUC > 0.7$	**91.2**	83.2	57.2	59.4	88.4	81.6
$AUC > 0.9$	27.0	**28.0**	21.8	18.4	21.6	21.0

	Gene expression splits (simulation)					
Avg. assays tested: 232.0	MO	MO-BC	GE	GE-S	CS-GC	CS-MF
Mean AUPRC	**0.263**	0.248	0.222	0.201	0.246	0.244
Mean AUROC	**0.664**	0.642	0.577	0.561	0.647	0.658
$AUC > 0.5$	155.6	150.2	127.6	127.2	153.2	**157.4**
$AUC > 0.7$	94.4	86.2	45.4	46.6	94.2	**99**
$AUC > 0.9$	**27.4**	23.6	14.2	12.6	22.6	22.4

	Morphology(bc)-based splits (simulation)					
Avg. assays tested: 179.8	MO	MO-BC	GE	GE-S	CS-GC	CS-MF
Mean AUPRC	0.224	0.207	0.199	0.198	0.225	**0.245**
Mean AUROC	0.634	0.600	0.562	0.564	0.631	**0.652**
$AUC > 0.5$	142	128.6	125.4	126.2	140.8	**143.6**
$AUC > 0.7$	**72.8**	63.0	49.2	49.2	81.0	82.6
$AUC > 0.9$	**21.6**	17.0	14.4	13.6	19.4	22.6

	Random splits (simulation)					
Avg. assays tested: 232.4	MO	MO-BC	GE	GE-S	CS-GC	CS-MF
Mean AUPRC	**0.259**	0.247	0.234	0.228	0.244	0.242
Mean AUROC	**0.670**	0.643	0.601	0.595	0.659	0.651
$AUC > 0.5$	**163.6**	154.2	145.6	144.0	157.6	157.8
$AUC > 0.7$	**97.2**	88.4	61.8	66.0	94.8	94.0
$AUC > 0.9$	**26.2**	22.0	20.4	17.4	25.8	23.4

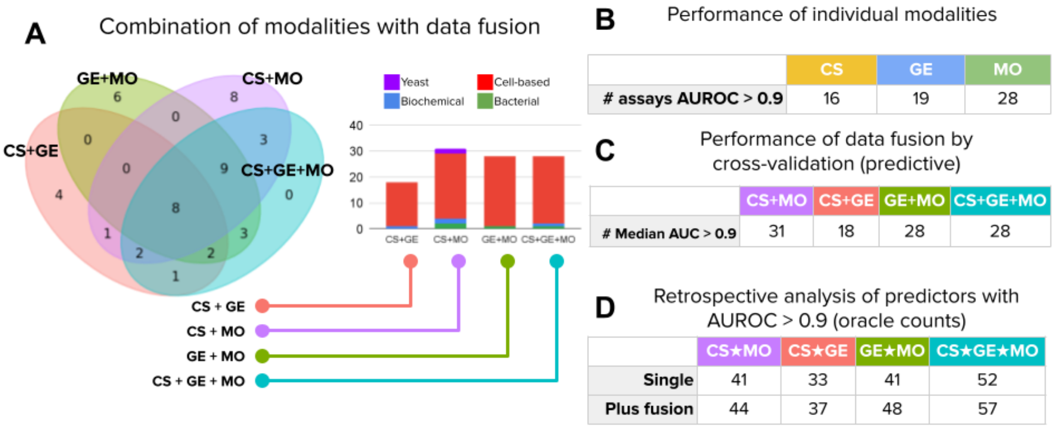
Table 1: Results of 5-fold cross-validation experiments. The tables present the mean results of 5-fold cross-validation experiments according to different data partition approaches. The metrics are: Mean AUPRC for 5 splits, Mean AUROC for 5 splits, mean counts of the predicted assays thresholded by AUROC ($AUC > 0.5$, $AUC > 0.7$, $AUC > 0.9$) for 5 splits. Sources of data used: MO: morphological features without batch-correction. MO-BC: morphological features with batch-correction. GE: Gene expression features. CS-GC: graph convolutional (GC) features. CS-MF: Morgan fingerprints. An average number of assays in the test set differs between modalities, as it is impossible to evaluate an assay without hits in the test set (which are different as different train-test split approaches were used). The source of the table [97].

	Baseline: independent modalities (scaffold-based partitions)					
	MO		GE		CS	
	Mean	Std	Mean	Std	Mean	Std
Mean AUPRC	0.252	0.021	0.234	0.038	0.232	0.036
Mean AUROC	0.637	0.021	0.592	0.034	0.630	0.018
$AUC > 0.5$	151.4	13.502	139.2	13.773	150.2	13.255
$AUC > 0.7$	83.2	11.100	57.2	16.316	88.4	6.066
$AUC > 0.9$	28.0	4.848	21.8	8.198	21.6	6.229

	Early fusion — concatenation (scaffold-based partitions)							
	GE-MO		MO-CS		GE-CS		GE-MO-CS	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Mean AUPRC	0.214	0.045	0.251	0.021	0.219	0.028	0.221	0.021
Mean AUROC	0.586	0.038	0.632	0.031	0.577	0.061	0.582	0.038
$AUC > 0.5$	138.8	18.377	151.8	19.905	138.6	26.773	137.2	22.928
$AUC > 0.7$	59.2	12.215	87.8	15.531	63.4	21.663	59.8	14.516
$AUC > 0.9$	16.0	4.743	23.6	4.159	17.0	2.292	20.4	4.278

	Late fusion — max pooling (scaffold-based partitions)							
	GE-MO		MO-CS		GE-CS		GE-MO-CS	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Mean AUPRC	0.261	0.026	0.267	0.034	0.251	0.039	0.265	0.032
Mean AUROC	0.652	0.028	0.661	0.027	0.645	0.026	0.665	0.031
$AUC > 0.5$	157.4	11.845	157.8	13.773	155.6	16.637	159.0	15.017
$AUC > 0.7$	86.0	9.670	98.8	7.430	87.0	9.566	96.4	10.877
$AUC > 0.9$	29.4	6.618	29.4	5.128	23.8	8.843	28.0	5.148

Table 2: Performance of individual and combined modalities for models trained with scaffold-based splits. The metrics are: Mean AUPRC for 5 splits, Mean AUROC for 5 splits, mean counts of the predicted assays thresholded by AUROC ($AUC > 0.5$, $AUC > 0.7$, $AUC > 0.9$) for 5 splits. Standard deviations are in a separate column. The source of the table [97].



Performance of individual modalities

	CS	GE	MO
# assays AUROC > 0.9	16	19	28

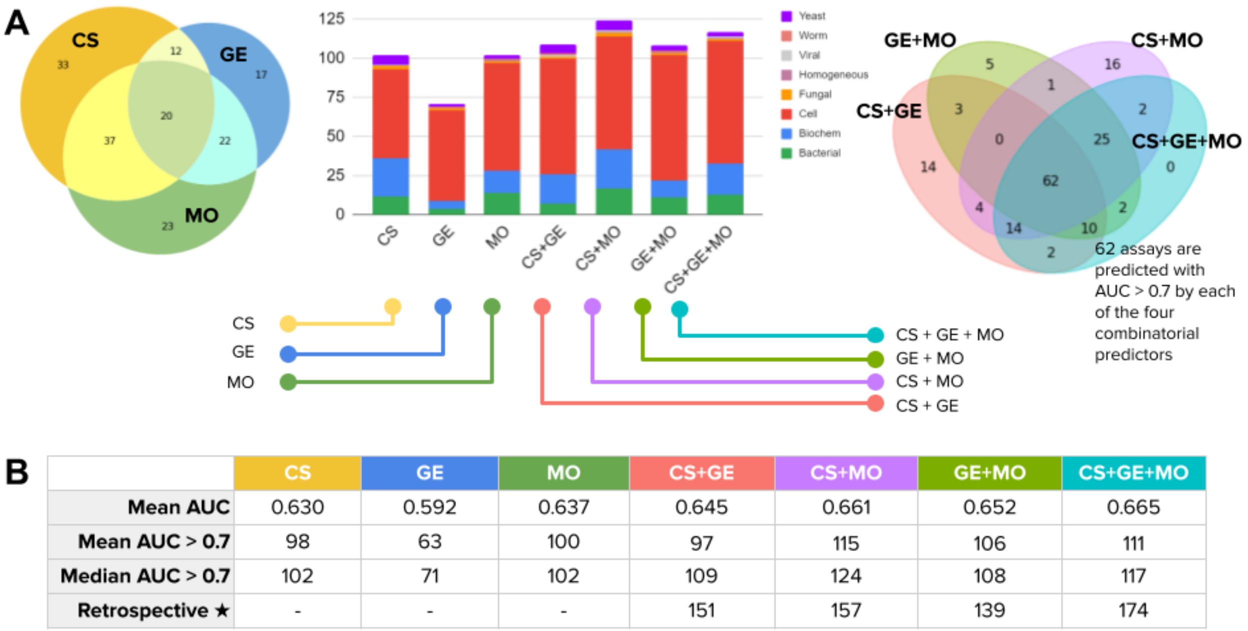
Performance of data fusion by cross-validation (predictive)

	CS+MO	CS+GE	GE+MO	CS+GE+MO
# Median AUC > 0.9	31	18	28	28

Retrospective analysis of predictors with AUROC > 0.9 (oracle counts)

	CS★MO	CS★GE	GE★MO	CS★GE★MO
Single	41	33	41	52
Plus fusion	44	37	48	57

Figure 32: Accurately predicted assays (median AUROC over splits is higher than 0.9). A. Venn diagram of accurately predicted assays using late fusion (left), bar plots show the distribution of accurately predicted assay types with late fusion (right). B. Number of accurately predicted assays per individual modality. C. Number of accurately predicted assays for combined modalities with the use of late fusion. Counts for median and mean AUROC over splits. D. Number of accurately predicted assays for retrospective analysis. "Single" is a simple union of the accurately predicted assays with individual modalities. "Plus fusion" is a union of accurately predicted assays with individual modalities plus the combined late fusion predictor. The source of the figure [97].



62 assays are predicted with AUC > 0.7 by each of the four combinatorial predictors

	CS	GE	MO	CS+GE	CS+MO	GE+MO	CS+GE+MO
Mean AUC	0.630	0.592	0.637	0.645	0.661	0.652	0.665
Mean AUC > 0.7	98	63	100	97	115	106	111
Median AUC > 0.7	102	71	102	109	124	108	117
Retrospective ★	-	-	-	151	157	139	174

Figure 33: Predicted assays with moderate accuracy (median AUROC over splits is higher than 0.7). A. Venn diagram of predicted assays with individual modalities (left), bar plot of predicted assay types by individual modalities and late fusion (center), Venn diagram of predicted assays with late fusion (right). B. Performance of individual modalities and late fusion. The metrics are: Mean AUC for 5 splits, mean counts of the predicted assays thresholded by AUROC ($AUC > 0.7$) for 5 splits. The source of the figure [97].

	CS	GE	MO	CS+GE	CS+MO	GE+MO	CS+GE+MO	Evaluated assays
Cell-based	7.05%	11.54%	13.46%	10.90%	16.03%	17.31%	16.67%	156
Biochemical	6.78%	0.00%	1.69%	1.69%	3.39%	0.00%	1.69%	59
Bacterial	0.00%	3.33%	16.67%	0.00%	6.67%	3.33%	3.33%	30
Yeast	5.56%	0.00%	5.56%	0.00%	11.11%	0.00%	0.00%	18
Fungal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3
Viral	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2
Worm	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1
Homogeneous	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1

Table 3: Predicted assays by type at the 0.9 threshold, median AUROC over scaffold-based splits was used. The source of the table [97].

	CS	GE	MO	CS+GE	CS+MO	GE+MO	CS+GE+MO	Evaluated assays
Cell-based	36.54%	37.18%	44.23%	47.44%	46.15%	51.28%	50.00%	156
Biochemical	40.68%	8.47%	23.73%	32.20%	42.37%	18.64%	33.90%	59
Bacterial	40.00%	13.33%	46.67%	23.33%	56.67%	36.67%	43.33%	30
Yeast	33.33%	11.11%	11.11%	33.33%	33.33%	16.67%	16.67%	18
Fungal	66.67%	33.33%	33.33%	33.33%	66.67%	33.33%	33.33%	3
Viral	50.00%	0.00%	0.00%	50.00%	50.00%	0.00%	50.00%	2
Worm	0.00%	100.00%	100.00%	100.00%	0.00%	100.00%	0.00%	1
Homogeneous	0.00%	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	1

Table 4: Predicted assays by type at the 0.7 threshold, median AUROC over scaffold-based splits was used. The source of the table [97].

4 Conclusions

The progress in computation and high-throughput biology methods is a mutual exchange: the rise of computational power paved the way for high-throughput methods. This, in turn, engages the computational powers by producing new piles of data, which have to be analyzed. As a part of those processes, new scientific sub-fields and computational analysis methods emerged. Imaging waited its turn, strengthening its methods from the wet lab and computational sides for a little while, even though the first image analysis attempts were successful and founded a new field [9] [19].

Biological image analysis skyrocketed in the middle of the 2010s when the shift from classical image analysis to deep-learning-based image analysis started and GPU-computation has become affordable. By this time, the wet-lab protocols for imaging were mostly established, and new specific protocols [10] and techniques (i.e. super-resolution) [107] appeared. The methods for image classification, detection and segmentation were swiftly adopted by the community of computational biologists for the specific tasks [7][108].

The sub-field of cell (nucleus) segmentation has matured in the last few years, besides the new methods (including attempts to build a general cell segmentation method) and additional post-processing methods, also new large-scale datasets and annotation tools were published [7]. Currently, new methods, usually specific for a particular domain of data are developed, but the community strives for general segmentation models and 3D segmentation [7].

As a part of the renaissance of phenotypic drug discovery [46], one of the biological imaging analysis sub-fields of particular interest with wide applicability of deep-learning methods [48] [51] is image-based profiling [10]. It is expected to advance in near future from both biological and computational sides [52]. From the computational side, all eyes are on unsupervised deep-learning methods. The hope is, that those will be more capable of capturing biologically relevant features of single-cells [93], rather than their supervised and weakly-supervised counterparts.

This thesis is focused on the usage of deep learning-based methods for single-cell segmentation and phenotypic profiling. From the segmentation side, the thesis presents the review of the nucleus segmentation sub-field, an annotation tool to create cell(nucleus) segmentation datasets and an evaluation of a post-processing method for nucleus segmentation. From the phenotyping side, the thesis presents weakly-supervised learning for large-scale image-based profiling and an evaluation of the predictive power of different cellular data modalities.

1. In a review paper, descriptions of the deep learning-based segmentation methods for 2D and 3D data, descriptions of the datasets and annotation tools. Several important points regarding the current state of the field of nuclei segmentation were expressed with the hope that the community will take those into account. The decision support helper tool for segmentation method selection was developed.

2. AnnotatorJ, the plugin for the popular imaging software ImageJ/Fiji, which utilizes

pre-trained models based on U-Net to ease the annotations of nuclei images. The experiments with expert annotators showed that AnnotatorJ reduces the time needed for the annotation and improves the accuracy of the produced annotations.

3. The test-time augmentation approach was experimentally evaluated for two popular deep learning frameworks: U-Net and Mask R-CNN. According to the observed results, it is possible to obtain additional segmentation accuracy with TTA on average, though in individual cases it is not guaranteed. Besides, in cases with underfit models, the usage of TTA marginally hurts the average segmentation performance. Visual observation of the images also showed, that TTA mostly modifies the output segmentations in the objects' borders, though in rare cases, especially in the case of Mask R-CNN, as it is instance segmentation-based the segmentations of the whole objects (improving segmentation by removing false positives or adding true positives). The recommendation would be to use TTA for the analysis of uncertain regions in segmentation. Besides, the computational cost of predictions increases with the use of TTA, but it is a concern only at a very large scale or if the inference is running on a CPU.

4. CNNs trained with a weakly-supervised learning approach were benchmarked in three large-scale profiling datasets versus classical features and pre-trained CNN baselines. The main finding is that by maximizing technical and phenotypic variation, WSL improves in capturing the biologically relevant representations. Batch-correction turned out to be a crucial element in capturing phenotypic variation. During this project, the combined Cell Painting dataset was gathered and a software tool DeepProfiler for deep learning-based image profiling was developed. As a result of experiments, a trained model for feature extraction from Cell Painting data was obtained.

5. The predictive power of different data modalities was evaluated: morphology, transcriptional profiles and chemical structures for the prediction of assay readouts. The results show that those three modalities individually can predict 6-10% of assays with high accuracy. According to experiments, those modalities turned out to be complementary combined and can provide up to 21% of assays that can be predicted with high accuracy or up to 64% if lower accuracy is acceptable.

List of Figures

List of Tables

References

[1] Peter Horvath, Nathalie Aulner, Marc Bickle, Anthony M Davies, Elaine Del Nery, Daniel Ebner, Maria C Montoya, Päivi Östling, Vilja Pietiäinen, Leo S Price, Spencer L Shorte, Gerardo Turcatti, Carina von Schantz, and Neil O Carragher. Screening out irrelevant cell-based models of disease. *Nat. Rev. Drug Discov.*, 15(11):751–769, November 2016.

[2] Ehud Shapiro, Tamir Biezuner, and Sten Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat. Rev. Genet.*, 14(9):618–630, September 2013.

[3] Lukas Badertscher, Thomas Wild, Christian Montellese, Leila T Alexander, Lukas Bammert, Marie Sarazova, Michael Stebler, Gabor Csucs, Thomas U Mayer, Nicola Zamboni, Ivo Zemp, Peter Horvath, and Ulrike Kutay. Genome-wide RNAi screening identifies protein modules required for 40S subunit synthesis in human cells. *Cell Rep.*, 13(12):2879–2891, December 2015.

[4] Olaf Wolkenhauer, Peter Wellstead, Kwang-Hyun Cho, Dhanya Mullassery, Caroline A Horton, Christopher D Wood, and Michael R H White. Single live-cell imaging for systems biology 9. *Essays Biochem.*, 45:121–134, September 2008.

[5] Jeffrey M Levsky, Shailesh M Shenoy, Rossanna C Pezo, and Robert H Singer. Single-cell gene expression profiling. *Science*, 297(5582):836–840, August 2002.

[6] Nikolai Slavov. Single-cell protein analysis by mass spectrometry. *Curr. Opin. Chem. Biol.*, 60:1–9, February 2021.

[7] Reka Hollandi, Nikita Moshkov, Lassi Paavolainen, Ervin Tasnadi, Filippo Piccinini, and Peter Horvath. Nucleus segmentation: towards automated solutions. *Trends Cell Biol.*, January 2022.

[8] Ben T Grys, Dara S Lo, Nil Sahin, Oren Z Kraus, Quaid Morris, Charles Boone, and Brenda J Andrews. Machine learning and computer vision approaches for phenotypic profiling. *J. Cell Biol.*, 216(1):65–71, January 2017.

[9] Zachary E. Perlman, Michael D. Slack, Yan Feng, Timothy J. Mitchison, Lani F. Wu, and Steven J. Altschuler. Multidimensional drug profiling by automated microscopy. *Science*, 306(5699):1194–1198, 2004.

[10] Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. January 2016.

[11] Juan C Caicedo, Allen Goodman, Kyle W Karhohs, Beth A Cimini, Jeanelle Ackerman, Marzieh Haghighi, Cherkeng Heng, Tim Becker, Minh Doan, Claire McQuin, Mohammad Rohban, Shantanu Singh, and Anne E Carpenter. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nat. Methods*, 16(12):1247–1253, December 2019.

[12] Réka Hollandi, Ákos Diósdi, Gábor Hollandi, Nikita Moshkov, and Péter Horváth. AnnotatorJ: an ImageJ plugin to ease hand annotation of cellular compartments. *Mol. Biol. Cell*, 31(20):2179–2186, September 2020.

[13] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nat. Methods*, 9(7):676–682, June 2012.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for Large-Scale image recognition. September 2014.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[16] Mohammad H Rohban, Hamdah S Abbasi, Shantanu Singh, and Anne E Carpenter. Capturing single-cell heterogeneity via data fusion improves image-based profiling. *Nat. Commun.*, 10(1):2082, May 2019.

[17] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.*, 59(8):3370–3388, August 2019.

[18] Aravind Subramanian, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, David L Lahr, Jodi E Hirschman, Zihan Liu, Melanie Donahue, Bina Julian, Mariya Khan, David Wadden, Ian C Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M Enache, Federica Piccioni, Sarah A Johnson, Nicholas J Lyons, Alice H Berger, Alykhan F Shamji, Angela N Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Y Takeda, Roger Hu, Desiree Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Peyton Greenside, Nathanael S Gray, Paul A Clemons, Serena Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J Haggarty, Lucienne V Ronco, Jesse S Boehm, Stuart L Schreiber, John G Doench, Joshua A Bittker, David E Root, Bang Wong, and Todd R Golub.

A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, November 2017.

[19] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, Polina Golland, and David M Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.*, 7(10):R100, October 2006.

[20] Nikita Moshkov, Botond Mathe, Attila Kertesz-Farkas, Reka Hollandi, and Peter Horvath. Test-time augmentation for deep learning-based cell segmentation on microscopy images. *Sci. Rep.*, 10(1):5068, March 2020.

[21] Aditya Pratapa, Michael Doron, and Juan C Caicedo. Image-based cell phenotyping with deep learning. *Curr. Opin. Chem. Biol.*, 65:9–17, December 2021.

[22] Erik Meijering. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Process. Mag.*, 29(5):140–145, September 2012.

[23] Christoph Sommer, Christoph Straehle, Ullrich Köthe, and Fred A Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 230–233, March 2011.

[24] Pascal Bamford and Brian Lovell. Unsupervised cell nucleus segmentation with active contours. *Signal Processing*, 71(2):203–213, December 1998.

[25] Jozsef Moinar, Adam Istvan Szucs, Csaba Molnar, and Peter Horvath. Active contours for selective object segmentation. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, March 2016.

[26] Csaba Molnar, Ian H Jermyn, Zoltan Kato, Vesa Rahkama, Päivi Östling, Piia Mikkonen, Vilja Pietiäinen, and Peter Horvath. Accurate morphology preserving segmentation of overlapping cells based on active contours. *Sci. Rep.*, 6:32412, August 2016.

[27] Adrien Hallou, Hannah G Yevick, Bianca Dumitrascu, and Virginie Uhlmann. Deep learning for bioimage analysis in developmental biology. *Development*, 148(18), September 2021.

[28] Srinivas Niranj Chandrasekaran, Hugo Ceulemans, Justin D Boyd, and Anne E Carpenter. Image-based profiling for drug discovery: due for a machine-learning upgrade? *Nat. Rev. Drug Discov.*, 20(2):145–159, February 2021.

[29] Riddhiman Dhar, Alsu M Missarova, Ben Lehner, and Lucas B Carey. Single cell functional genomics reveals the importance of mitochondria in cell-to-cell phenotypic variation. *Elife*, 8, January 2019.

[30] Tomohiro Hayakawa, V B Surya Prasath, Hiroharu Kawanaka, Bruce J Aronow, and Shinji Tsuruoka. Computational nuclei segmentation methods in digital pathology: A survey. *Arch. Comput. Methods Eng.*, 28(1):1–13, January 2021.

[31] Nobuyuki Otsu. A threshold selection method from Gray-Level histograms. *IEEE Trans. Syst. Man Cybern.*, 9(1):62–66, January 1979.

[32] Stéfan van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and scikit-image contributors. scikit-image: image processing in python. *PeerJ*, 2:e453, June 2014.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 234–241. Springer International Publishing, Cham, 2015.

[34] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods*, 18(1):100–106, January 2021.

[35] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture notes in computer science, pages 265–273. Springer International Publishing, Cham, 2018.

[36] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017.

[37] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, Maria Kovacs, Ede Migh, Allen Goodman, Tamas Balassa, Krisztian Koos, Wenyu Wang, Juan Carlos Caicedo, Norbert Bara, Ferenc Kovacs, Lassi Paavolainen, Tivadar Danka, Andras Kriston, Anne Elizabeth Carpenter, Kevin Smith, and Peter Horvath. NucleAIzer: A parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Syst.*, 10(5):453–458.e6, May 2020.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[40] Andrew Y Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Twenty-first international conference on Machine learning - ICML '04*, New York, New York, USA, 2004. ACM Press.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, June 2017.

[43] Ellen L Berg. The future of phenotypic drug discovery. *Cell Chem Biol*, 28(3):424–430, March 2021.

[44] David C Swinney and Jonathan A Lee. Recent advances in phenotypic drug discovery. *F1000Res.*, 9, August 2020.

[45] Jörg Eder, Richard Sedrani, and Christian Wiesmann. The discovery of first-in-class drugs: origins and evolution. *Nat. Rev. Drug Discov.*, 13(8):577–587, August 2014.

[46] Fabien Vincent, Arsenio Nueda, Jonathan Lee, Monica Schenone, Marco Prunotto, and Mark Mercola. Phenotypic drug discovery: recent successes, lessons learned and new directions. *Nature Reviews Drug Discovery*, may 2022.

[47] Sigrun M Gustafsdottir, Vebjorn Ljosa, Katherine L Sokolnicki, J Anthony Wilson, Deepika Walpita, Melissa M Kemp, Kathleen Petri Seiler, Hyman A Carrel, Todd R Golub, Stuart L Schreiber, Paul A Clemons, Anne E Carpenter, and Alykhan F Shamji. Multiplex cytological profiling assay to measure diverse cellular states. *PLoS One*, 8(12):e80999, December 2013.

[48] Juan C Caicedo, Sam Cooper, Florian Heigwer, Scott Warchal, Peng Qiu, Csaba Molnar, Aliaksei S Vasilevich, Joseph D Barry, Harmanjit Singh Bansal, Oren Kraus, Mathias Wawer, Lassi Paavolainen, Markus D Herrmann, Mohammad Rohban, Jane Hung, Holger Hennig, John Concannon, Ian Smith, Paul A Clemons, Shantanu Singh, Paul Rees, Peter Horvath, Roger G Linington, and Anne E Carpenter. Data-analysis strategies for image-based cell profiling. *Nat. Methods*, 14(9):849–863, August 2017.

[49] Joseph Boyd. *Deep learning for computational phenotyping in cell-based assays.* PhD thesis, Université Paris sciences et lettres, June 2020.

[50] Juan C Caicedo, Claire McQuin, Allen Goodman, Shantanu Singh, and Anne E Carpenter. Weakly supervised learning of Single-Cell feature embeddings. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018:9309–9318, June 2018.

[51] Nick Pawlowski, Juan C Caicedo, Shantanu Singh, Anne E Carpenter, and Amos Storkey. Automating morphological profiling with generic deep convolutional networks. November 2016.

[52] Srinivas Niranj Chandrasekaran, Hugo Ceulemans, Justin D. Boyd, and Anne E. Carpenter. Image-based profiling for drug discovery: due for a machine-learning upgrade? *Nature Reviews Drug Discovery*, 20(2):145–159, dec 2020.

[53] Mingyue Zheng, Jihui Zhao, Chen Cui, Zunyun Fu, Xutong Li, Xiaohong Liu, Xiaoyu Ding, Xiaoqin Tan, Fei Li, Xiaomin Luo, Kaixian Chen, and Hualiang Jiang. Computational chemical biology and drug design: Facilitating protein structure, function, and modulation studies. *Med. Res. Rev.*, 38(3):914–950, May 2018.

[54] Douglas B Kell, Soumitra Samanta, and Neil Swainston. Deep learning and generative methods in cheminformatics and chemical biology: navigating small molecule space intelligently. *Biochem. J*, 477(23):4559–4580, December 2020.

[55] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, February 1988.

[56] B Zagidullin, Z Wang, Y Guan, E Pitkänen, and J Tang. Comparative analysis of molecular fingerprints in prediction of drug combination effects. *Brief. Bioinform.*, 22(6), November 2021.

[57] H L Morgan. The generation of a unique machine description for chemical Structures-A technique developed at chemical abstracts service. *J. Chem. Doc.*, 5(2):107–113, May 1965.

[58] G W Bemis and M A Murcko. The properties of known drugs. 1. molecular frameworks. *J. Med. Chem.*, 39(15):2887–2893, July 1996.

[59] Karren Yang, Samuel Goldman, Wengong Jin, Alex Lu, Regina Barzilay, Tommi Jaakkola, and Caroline Uhler. Improved conditional flow models for molecule to image synthesis. June 2020.

[60] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, Victoria M Tran, Anush Chiappino-Pepe, Ahmed H Badran, Ian W Andrews, Emma J Chory, George M Church, Eric D Brown, Tommi S Jaakkola, Regina Barzilay, and James J Collins. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13, February 2020.

[61] Kai Yao, Kaizhu Huang, Jie Sun, Linzhi Jing, Dejian Huang, and Curran Jude. Scaffold-A549: A benchmark 3D fluorescence image dataset for unsupervised nuclei segmentation. *Cognit. Comput.*, November 2021.

[62] Christoffer Edlund, Timothy R Jackson, Nabeel Khalid, Nicola Bevan, Timothy Dale, Andreas Dengel, Sheraz Ahmed, Johan Trygg, and Rickard Sjögren. LIVECell-A

large-scale dataset for label-free live cell segmentation. *Nat. Methods*, 18(9):1038–1045, September 2021.

[63] Florin C Walter, Sebastian Damrich, and Fred A Hamprecht. Multistar: Instance segmentation of overlapping objects with star-convex polygons. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. IEEE, April 2021.

[64] Soham Mandal and Virginie Uhlmann. Splinedist: Automated cell segmentation with spline curves. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1082–1086, April 2021.

[65] Linfeng Yang, Rajarshi P Ghosh, J Matthew Franklin, Simon Chen, Chenyu You, Raja R Narayan, Marc L Melcher, and Jan T Liphardt. NuSeT: A deep learning tool for reliably separating and analyzing crowded cells. *PLoS Comput. Biol.*, 16(9):e1008193, September 2020.

[66] Ulysse Rubens, Romain Mormont, Lassi Paavolainen, Volker Bäcker, Benjamin Pavie, Leandro A Scholz, Gino Michiels, Martin Maška, Devrim Ünay, Graeme Ball, Renaud Hoyoux, Rémy Vandaele, Ofra Golani, Stefan G Stanciu, Natasa Sladoje, Perrine Paul-Gilloteaux, Raphaël Marée, and Sébastien Tosi. BIAFLOWS: A collaborative framework to reproducibly deploy and benchmark bioimage analysis workflows. *Patterns (N Y)*, 1(3):100040, June 2020.

[67] Ruchika Verma, Neeraj Kumar, Abhijeet Patil, Nikhil Cherian Kurian, Swapnil Rane, Simon Graham, Quoc Dang Vu, Mieke Zwager, Shan E Ahmed Raza, Nasir Rajpoot, Xiyi Wu, Huai Chen, Yijie Huang, Lisheng Wang, Hyun Jung, G Thomas Brown, Yanling Liu, Shuolin Liu, Seyed Alireza Fatemi Jahromi, Ali Asghar Khani, Ehsan Montahaei, Mahdieh Soleymani Baghshah, Hamid Behroozi, Pavel Semkin, Alexandr Rassadin, Prasad Dutande, Romil Lodaya, Ujjwal Baid, Bhakti Baheti, Sanjay Talbar, Amirreza Mahbod, Rupert Ecker, Isabella Ellinger, Zhipeng Luo, Bin Dong, Zhengyu Xu, Yuehan Yao, Shuai Lv, Ming Feng, Kele Xu, Hasib Zunair, Abdessamad Ben Hamza, Steven Smiley, Tang-Kai Yin, Qi-Rui Fang, Shikhar Srivastava, Dwarikanath Mahapatra, Lubomira Trnavska, Hanyun Zhang, Priya Lakshmi Narayanan, Justin Law, Yinyin Yuan, Abhiroop Tejomay, Aditya Mitkari, Dinesh Koka, Vikas Ramachandra, Lata Kini, and Amit Sethi. MoNuSAC2020: A multi-organ nuclei segmentation and classification challenge. *IEEE Trans. Med. Imaging*, pages 1–1, 2021.

[68] Kazuhisa Matsunaga, Akira Hamada, Akane Minagawa, and Hiroshi Koga. Image classification of melanoma, nevus and seborrheic keratosis by deep neural network ensemble. March 2017.

[69] Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. April 2018.

[70] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks, 2019.

[71] Csilla Brasko, Kevin Smith, Csaba Molnar, Nora Farago, Lili Hegedus, Arpad Balind, Tamas Balassa, Abel Szkalisity, Farkas Sukosd, Katalin Kocsis, Balazs Balint, Lassi Paavolainen, Marton Z Enyedi, Istvan Nagy, Laszlo G Puskas, Lajos Haracska, Gabor Tamas, and Peter Horvath. Intelligent image-based in situ single-cell isolation, 2018.

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

[73] Luis Pedro Coelho, Aabid Shariff, and Robert F Murphy. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms, 2009.

[74] Kevin Smith, Yunpeng Li, Filippo Piccinini, Gabor Csucs, Csaba Balazs, Alessandro Bevilacqua, and Peter Horvath. CIDRE: an illumination-correction method for optical microscopy, 2015.

[75] Peter Naylor, Marick Lae, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map, 2019.

[76] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Trans. Med. Imaging*, 36(7):1550–1560, July 2017.

[77] Juan C. Caicedo, Jonathan Roth, Allen Goodman, Tim Becker, Kyle W. Karhohs, Matthieu Broisin, Csaba Molnar, Claire McQuin, Shantanu Singh, Fabian J. Theis, and Anne E. Carpenter. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, 95(9):952–965, 2019.

[78] zhixuhao. zhixuhao/unet. `https://github.com/zhixuhao/unet`. Accessed: 2019-10-7.

[79] Tensorflow Developers. TensorFlow, 2021.

[80] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. December 2014.

[81] matterport. matterport/Mask_RCNN. `https://github.com/matterport/Mask_RCNN`. Accessed: 2019-10-7.

[82] Nikita Moshkov, Michael Bornholdt, Santiago Benoit, Claire McQuin, Matthew Smith, Allen Goodman, Rebecca Senft, Yu Han, Mehrtash Babadi, Peter Horvath, Beth A. Cimini, Anne E. Carpenter, Shantanu Singh, and Juan C Caicedo. Learning representations for image-based profiling of perturbations. *bioRxiv*, 2022.

[83] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *J. Educ. Psychol.*, 66(5):688–701, October 1974.

[84] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In Maria Florina Balcan and Kilian Q Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 3020–3029, New York, New York, USA, 2016. PMLR.

[85] Mohammad Hossein Rohban, Shantanu Singh, Xiaoyun Wu, Julia B Berthet, Mark-Anthony Bray, Yashaswi Shrestha, Xaralabos Varelas, Jesse S Boehm, and Anne E Carpenter. Systematic morphological profiling of human gene and allele function via cell painting. *Elife*, 6, March 2017.

[86] Mark-Anthony Bray, Sigrun M Gustafsdottir, Mohammad H Rohban, Shantanu Singh, Vebjorn Ljosa, Katherine L Sokolnicki, Joshua A Bittker, Nicole E Bodycombe, Vlado Dancík, Thomas P Hasaka, Cindy S Hon, Melissa M Kemp, Kejie Li, Deepika Walpita, Mathias J Wawer, Todd R Golub, Stuart L Schreiber, Paul A Clemons, Alykhan F Shamji, and Anne E Carpenter. A dataset of images and morphological profiles of 30 000 small-molecule treatments using the cell painting assay. *Gigascience*, 6(12):1–5, December 2017.

[87] J C Caicedo, J Arevalo, F Piccioni, and others. Cell painting predicts impact of lung cancer variants. *Mol. Biol. Cell*, 2022.

[88] Gregory P Way, Ted Natoli, Adeniyi Adeboye, Lev Litichevskiy, Andrew Yang, Xiaodong Lu, Juan C Caicedo, Beth A Cimini, Kyle Karhohs, David J Logan, Mohammad Rohban, Maria Kost-Alimova, Kate Hartland, Michael Bornholdt, Niranj Chandrasekaran, Marzieh Haghighi, Shantanu Singh, Aravind Subramanian, and Anne E Carpenter. Morphology and gene expression profiling provide complementary information for mapping cell state. October 2021.

[89] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks. May 2019.

[90] Stanley Bryan Z Hua, Alex X Lu, and Alan M Moses. CytoImageNet: A large-scale pretraining dataset for bioimage transfer learning. November 2021.

[91] Vebjorn Ljosa, Peter D Caie, Rob Ter Horst, Katherine L Sokolnicki, Emma L Jenkins, Sandeep Daya, Mark E Roberts, Thouis R Jones, Shantanu Singh, Auguste Genovesio,

Paul A Clemons, Neil O Carragher, and Anne E Carpenter. Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. *J. Biomol. Screen.*, 18(10):1321–1329, December 2013.

[92] D Michael Ando, Cory Y McLean, and Marc Berndl. Improving phenotypic measurements in High-Content imaging screens. July 2017.

[93] Alexis Perakis, Ali Gorji, Samriddhi Jain, Krishna Chaitanya, Simone Rizza, and Ender Konukoglu. Contrastive learning of Single-Cell phenotypic representations for treatment classification. In *Machine Learning in Medical Imaging*, pages 565–575. Springer International Publishing, 2021.

[94] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *Am. Stat.*, 72(4):309–314, October 2018.

[95] Christopher D Manning. *Introduction to information retrieval.* Syngress Publishing,, 2008.

[96] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W H Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.*, December 2018.

[97] Nikita Moshkov, Tim Becker, Kevin Yang, Peter Horvath, Vlado C Dancik, Bridget K Wagner, Paul C Clemons, Shantanu Singh, Anne E Carpenter, and Juan C Caicedo. Predicting compound activity from phenotypic profiles and chemical structures. December 2020.

[98] M Hofmarcher, E Rumetshofer, D A Clevert, and others. Accurate prediction of biological assays with high-throughput microscopy images and convolutional networks. *Journal of chemical*, 2019.

[99] Gregory P Way, Maria Kost-Alimova, Tsukasa Shibue, William F Harrington, Stanley Gill, Federica Piccioni, Tim Becker, William C Hahn, Anne E Carpenter, Francisca Vazquez, and Shantanu Singh. Predicting cell health phenotypes using image-based morphology profiling. July 2020.

[100] Jonathan B Baell and Georgina A Holloway. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.*, 53(7):2719–2740, April 2010.

[101] Mathias J Wawer, David E Jaramillo, Vlado Dančík, Daniel M Fass, Stephen J Haggarty, Alykhan F Shamji, Bridget K Wagner, Stuart L Schreiber, and Paul A Clemons. Automated Structure-Activity relationship mining: Connecting chemical structure to biological profiles. *J. Biomol. Screen.*, 19(5):738–748, June 2014.

[102] Mathias J Wawer, Kejie Li, Sigrun M Gustafsdottir, Vebjorn Ljosa, Nicole E Bodycombe, Melissa A Marton, Katherine L Sokolnicki, Mark-Anthony Bray, Melissa M Kemp, Ellen Winchester, Bradley Taylor, George B Grant, C Suk-Yee Hon, Jeremy R Duvall, J Anthony Wilson, Joshua A Bittker, Vlado Dančík, Rajiv Narayan, Aravind Subramanian, Wendy Winckler, Todd R Golub, Anne E Carpenter, Alykhan F Shamji, Stuart L Schreiber, and Paul A Clemons. Toward performance-diverse small-molecule libraries for cell-based phenotypic screening using multiplexed high-dimensional profiling. *Proceedings of the National Academy of Sciences*, 111(30):10911–10916, July 2014.

[103] Sebastian G Rohrer and Knut Baumann. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *J. Chem. Inf. Model.*, 49(2):169–184, February 2009.

[104] jingw. GitHub - jingw2/size_constrained_clustering: Implementation of size constrained clustering algorithm. `https://github.com/jingw2/size_constrained_clustering`. Accessed: 2022-4-3.

[105] Jaak Simm, Günter Klambauer, Adam Arany, Marvin Steijaert, Jörg Kurt Wegner, Emmanuel Gustin, Vladimir Chupakhin, Yolanda T Chong, Jorge Vialard, Peter Buijnsters, Ingrid Velter, Alexander Vapirev, Shantanu Singh, Anne E Carpenter, Roel Wuyts, Sepp Hochreiter, Yves Moreau, and Hugo Ceulemans. Repurposing High-Throughput image assays enables biological activity prediction for drug discovery. *Cell Chem Biol*, 25(5):611–618.e3, May 2018.

[106] Maris Lapins and Ola Spjuth. Evaluation of gene expression and phenotypic profiling data as quantitative descriptors for predicting drug targets and mechanisms of action. July 2019.

[107] Malte Renz. Fluorescence microscopy—a historical and technical perspective. *Cytometry Part A*, 83(9):767–779, 2013.

[108] Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. Deep learning for cellular image analysis. *Nature Methods*, 16(12):1233–1246, may 2019.