CellPress

Cell Systems
Review

Phenotypic Image Analysis Software Tools for Exploring and Understanding Big Image Data from Cell-Based Assays

Kevin Smith,[1,2] Filippo Piccinini,[3] Tamas Balassa,[4] Krisztian Koos,[4] Tivadar Danka,[4] Hossein Azizpour,[1,2] and Peter Horvath[4,5,*]

[1]KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, Lindstedtsvägen 3, 10044 Stockholm, Sweden
[2]Science for Life Laboratory, Tomtebodavägen 23A, 17165 Solna, Sweden
[3]Istituto Scientifico Romagnolo per lo Studio e la Cura dei Tumori (IRST) IRCCS, Via P. Maroncelli 40, Meldola, FC 47014, Italy
[4]Synthetic and Systems Biology Unit, Hungarian Academy of Sciences, Biological Research Center (BRC), Temesvári krt. 62, 6726 Szeged, Hungary
[5]Institute for Molecular Medicine Finland, University of Helsinki, Tukholmankatu 8, 00014 Helsinki, Finland
*Correspondence: horvath.peter@brc.mta.hu
https://doi.org/10.1016/j.cels.2018.06.001

Phenotypic image analysis is the task of recognizing variations in cell properties using microscopic image data. These variations, produced through a complex web of interactions between genes and the environment, may hold the key to uncover important biological phenomena or to understand the response to a drug candidate. Today, phenotypic analysis is rarely performed completely by hand. The abundance of high-dimensional image data produced by modern high-throughput microscopes necessitates computational solutions. Over the past decade, a number of software tools have been developed to address this need. They use statistical learning methods to infer relationships between a cell's phenotype and data from the image. In this review, we examine the strengths and weaknesses of non-commercial phenotypic image analysis software, cover recent developments in the field, identify challenges, and give a perspective on future possibilities.

Introduction

One of the greatest scientific achievements of the past century is the complete sequencing of the human genome (Venter and Zhu, 2001). This ambitious project in genomics was a "moonshot," a ground-breaking exploratory undertaking with unforeseeable risks and benefits. Today, many believe that the next great challenge in biology lays in "*phenomics*"—the quantification of the set of phenotypes that completely characterizes an organism (the "phenome") (Houle et al., 2010; Rozenblatt-Rosen et al., 2017). A phenotype is defined as the set of observable characteristics of an organism—its morphology, biochemical properties, behavior, etc. By collecting and analyzing rich phenotypic data, we hope to improve our understanding of how genetic and environmental factors give rise to changes in organisms or in their behavior, and to better predict important outcomes such as fitness, reproduction, crop yield, disease, cancerogenesis, resistance, or mortality. In contrast to the genome, a complete understanding of the phenome is not possible with current technology. The complexity and information content of the phenome is vastly greater than that of the genome—therefore it is crucial that in our quest to understand the phenome, we intelligently choose what to measure and which phenomics tools to use.

Imaging is a powerful and highly flexible technology for studying phenomics. It can capture spatial and temporal information with high fidelity from the nanometer scale to the whole organism. It implicitly represents the morphological characteristics of the cell, and, using labeling technologies such as fluorescent tags, it is possible to localize subcellular structures, proteins, and other molecules. Images are often cheap and quick to acquire, allowing us to conduct large-scale screening experiments that, for example, visualize and test phenotypic responses to genetic perturbations or drug treatments. Recent advances in microscopy, automation, and computation have dramatically increased our ability to generate images rich in phenotypic information, and, today, images are generated orders of magnitude faster than they can be manually inspected. Consequently, to obtain phenotypic readouts we have come to rely on "phenotypic image analysis" techniques—computational methods that transform raw image data into useful phenotypic measurements.

The aim of this review is to provide an overview of phenotypic image analysis for cell-based image assays—its origins, existing software solutions, challenges, and perspectives for the future. We begin with some background including a brief history of the field, and then provide a critical summary of the predominant free and open-source tools for performing phenotypic analysis. We then survey some recent trends in phenotypic analysis, and discuss some of the main challenges. Finally, we speculate about future directions that research in phenotypic image analysis might take.

There are two important remarks regarding the scope of this review. First, we concentrate on high-level methods for interpretation and analysis of phenotypic image data, avoiding intermediate tasks such as image processing, normalization, or segmentation (Lucchi et al., 2015). Caicedo et al. (2017) provide a thorough review of these techniques. A second point is that this article focuses on cell-based assays. Phenotypic profiling is applied to other types of imaging data including whole

organisms or tissues sections. Some specialized tools exist for these types of experiments (Robertson et al., 2017), but they are beyond the scope of this article.

A History of Phenotypic Image Analysis in Cell-Based Assays

Fully automated high-throughput screening (HTS) assays allow researchers to quickly conduct millions of chemical, genetic, and pharmacological tests to rapidly identify active compounds, antibodies, or genes that affect a particular biological process. Some of the first non-automated screens to classify mutations based on similar phenotypes were performed in the late 1970s (Brenner, 1974; Nüsslein-Volhard and Wieschaus, 1980). Soon after, advances in automation enabled early high-throughput drug discovery screens in the 1980s (Pereira and Williams, 2007). Since then, HTS has been widely adopted in academic and pharmaceutical research, especially for studying effects of anticancer drugs. However, readouts from HTS are usually univariate, their resolution is often limited to the well level, and many experiments are not suited to conventional HTS methods (Thomsen et al., 2005).

High-content analysis applies image analysis methods to automate cellular measurements, including the quantification of cellular products such as proteins, or detection of changes in morphology (Zanella et al., 2010). High-content screening (HCS) is the combination of high-content analysis and HTS (Bickle, 2010), offering richer data, higher throughput, and increased flexibility (Usaj et al., 2016). One of the first studies to employ a high-content screen aimed to measure drug-induced transport of a GFP-tagged human glucocorticoid receptor chimeric protein in tumor cells (Giuliano et al., 1997). In the years that followed, crucial developments toward automated high-throughput microscopy, such as camera autofocus, plate/sample positioning, and high-density formats have made HCS a viable strategy (Rimon and Schuldiner, 2011). HCS has since become an important tool for detecting changes in fluorescent reporter genes (Chia et al., 2010; Desbordes et al., 2008), subcellular localization of proteins (Orvedahl et al., 2011; Link et al., 2009), inferring biological pathways (Rämö et al., 2014), inferring the mechanism of action of small molecules (Young et al., 2008) and classifying drugs (Loo et al., 2007). For a review of applications and approaches related to HCS, see (Boutros et al., 2015; Usaj et al., 2016; Singh et al., 2014; Zanella et al., 2010).

Starting around 2006, several free and open-source image analysis software solutions have been released to the research community. CellProfiler, a software package developed at the Broad Institute, allows users to mix and match modules to create their own customized image analysis pipelines (Carpenter et al., 2006). CellProfiler quickly gained popularity because it put flexible and powerful high-content image analysis (Eliceiri et al., 2012) into the hands of a wide group of researchers, even those without extensive programming skills (Kamentsky et al., 2011). BioConductor (Huber et al., 2015), with the EBImage (Pau et al., 2010) and imageHTS (Pau et al., 2018) packages, allows users to create custom image analysis pipelines in the R programming language to segment cells and extract features. Other open-source image analysis tools include Icy (De Chaumont et al., 2012), BioImageXD (Kankaanpää et al., 2012), and ImageJ/Fiji (Schindelin et al., 2012). Pharmaceutical companies have widely adopted these tools in their assays because of their flexibility and ability to scale to large-scale experiments using cluster or cloud computing. For live-cell imaging, software solutions include the ADAPT plugin for ImageJ (Barry et al., 2015) and NeuriteTracker (Fusco et al., 2016). For a comprehensive review of open-source software solutions for segmenting and quantifying microscopy images, see (Eliceiri et al., 2012; Shamir et al., 2010; Sommer and Gerlich, 2013). Commercial image analysis software offer alternative solutions, and are often provided with HCS systems from microscopy companies. Although these tools are sufficient for many routine HCS experiments, they can be difficult to extend or customize for specific assays.

As our ability to perform basic processing on phenotypic image data has increased, the problem of how to make sense of the data has come to the forefront. Many studies resort to using only a single measurement or a small set of measurements to explain phenotypic behavior. The measurements and associated thresholds used to categorize cells are often hand-selected or designed, sometimes with considerable effort. While this approach provides straightforward explanations of the analysis, some have criticized it for over-simplification and a failure to utilize the full potential of the data. The evidence suggests that the analysis suffers as a result (Singh et al., 2014).

Image-based screens routinely contain millions of cells, and modern image analysis software is capable of extracting hundreds or even thousands of features to describe each cell. Humans cannot be reasonably expected to select the optimal measurements to interpret so many parameters. However, machine-learning methods possess this capability. By providing examples of cells with corresponding annotations describing their phenotype, machine-learning methods can learn patterns in the data that are most predictive of cellular phenotype. Models trained this way can reliably predict the phenotype for cells they have never seen before. This machine-learning-based paradigm to phenotypic profiling has become increasingly popular in recent years.

Machine-Learning Concepts for Phenotypic Analysis

Before introducing the software tools for phenotypic analysis, we give a brief overview of machine learning and how it is applied for phenotypic analysis. A glossary of commonly used machine-learning terminology is provided in Box 1. Machine learning involves the development of theories and techniques to automatically learn to perform a task from data. There are various types of tasks, the most common of which are categorization of samples into certain classes (“classification”), grouping of similar samples (“clustering”), and estimation of real values from data (“regression”). A machine-learning system gets an input sample, processes it, and produces some desired form of prediction; a category or value. In phenotypic image analysis, it is common for the input sample to be an image of a cell, and for the desired output to be the phenotypic category of the cell.

For the machine-learning algorithm to learn how to process an input sample and produce the desired output, it requires a training dataset. A training set is a collection of input samples acquired for the purpose of learning (e.g., a set of 1,000 cell images with various phenotypes). There are different categories of learning that can be performed on the data. In supervised learning, in order to learn how to process the input sample, we annotate the training samples with their correct output (also known as “label”). Thus, the learning algorithm needs to find

Box 1. Machine-Learning Terms

ACCURACY

A measure of the performance of a machine-learning system. It reflects how close the predictions of the system are to the actual measurements (annotations) in a test scenario. For classification tasks, it is usually calculated by the ratio of correctly classified samples to the total number of samples and is indicated by percentage values.

ACTIVE LEARNING

A learning process that involves obtaining new training samples while learning. The learning system, at several iterations during learning, chooses which samples should be annotated and added to the training set, thus it is an active process. Usually, the learning system chooses new samples based on their informativeness. This makes the model efficient in the required amount of data and minimizes the annotation efforts.

ANNOTATION

Refers to the process of annotating individual datums with the desired labels, or to the labels themselves. For instance, one can annotate a dataset of cells into several phenotype class labels (the annotations). Annotated data can be used for training a supervised machine-learning system, and/or measuring its performance.

CLASS

The task of a machine-learning system can be to predict, for an input sample, a value from a discrete set. Each value in this discrete set is, then, called a class. For the task of phenotype recognition, each phenotype is considered a class.

CLASSIFICATION

The task of a machine-learning system can be to predict a value from a discrete set. This type of prediction tasks is called classification. Phenotype recognition is a classification task. If the task is to predict two classes (e.g., existence and nonexistence of a concept), it is called a binary classification. When the task is to detect more than two concepts it is a multi-class classification task. A multi-label classification task is when a sample can contain multiple classes at the same time (for example, presence of proteins within a cell).

CLASS IMBALANCE

Where the number of samples belonging to different classes are highly biased. Most training algorithms try to maximize the average performance over all samples. When one class has substantially more samples than another, the model will biased in detecting the larger class over the smaller, as it improves performance. There are different ways to mitigate class imbalance. One common method is to re-sample the training dataset to make the sample distribution over different classes uniform.

CLUSTERING

Tries to group samples into several clusters by maximizing the intra-cluster similarity and minimizing inter-cluster similarities. There are different similarity measures and learning algorithms proposed for clustering. Among the most commonly used algorithms are *k*-means, spectral, and agglomerative clustering, as well as density-based methods such as DBSCAN. Clustering is an unsupervised learning algorithm since it requires no annotation.

CROSS-VALIDATION

A technique for testing how well a model generalizes in the absence of a holdout test set. The training data are divided into a number of subsets, and then the same number of models is built, with each subset held out in turn. Each of those models is tested on the holdout sample, and the average accuracy of the models on those holdout samples is used to estimate the accuracy of the model when applied to new data.

DATA TYPES

When working with statistical methods including machine learning, it is important to understand the different types of data. Numerical data are expressed as numbers, and often have a meaning such as an intensity measurement. Numerical data can be discrete,

(Continued on next page)

Box 1. *Continued*

taking only integer values, or continuous taking any real number. Categorical data represent characteristics that are not expressed with numbers, such as phenotype. Categorical data can be ordinal if they can be ranked (e.g., the infection level of a cell is: 0, uninfect; 1, weakly infected; 2, strongly infected) or nominal if it cannot be ordered (e.g., cell line).

DEEP LEARNING

A machine-learning model which applies multiple layers of transformation to the input sample such that the final transformation in the last layer produces the desired output. Deep networks usually process the raw inputs, which puts them in contrast to traditional models that require hand-crafted representation of samples. The effectiveness of deep learning mainly comes from this fact: it learns the representation most suitable for the data.

ENSEMBLE LEARNING

Helps decreasing the variance of a machine-learning model by combining different models into an ensemble. Models can overfit to the training data, which inhibits the generalization of the model to unseen samples. Making an ensemble of models each of which is trained on different data and/or with different settings can help alleviate this issue.

FEATURE

The input to a machine-learning model is a set of values. Each of these values is usually called a feature, and the whole set is called a feature representation or simply representation. These features can be the raw input (e.g., pixel intensities for an input image) or hand-crafted measurement (e.g., some human-designed filters applied to an image).

FEATURE SELECTION

The selection of a subset of the feature set that is informative for predictive modeling. Feature selection is related to dimensionality reduction in that it seeks to remove noisy signals from the input data. Feature selection may select features based on performance and does not modify the features, in contrast to dimensionality reduction methods such as principal-component analysis.

GENERALIZATION

The applicability of a learned model to unseen samples is called generalization. In an ideal case of generalization, a model's error does not increase when going from the training set to unseen samples (e.g., the held-out test set). When there is a large gap between the training and test error, the model is experiencing "overfitting."

INFERENCE

Refers to the process of obtaining the prediction of the model for an input sample.

INSTANCE

A single instance of our data, an input sample.

OVERFITTING

A model that does not generalize well to real-world cases although it fits the training data well is overfitting. A common explanation is that the model has "memorized" the training data rather than understanding the fundamental concepts for the task.

PARAMETER (AND PARAMETER TUNING)

Degrees of freedom of a model. Some parameters of a machine-learning model are automatically learnable from training data and using machine-learning algorithms. Some other parameters of a model are for the machine-learning experts to select for each task individually. The latter are usually referred to as the hyper-parameters of the model, and the process of selecting them is called (hyper) parameter tuning.

(Continued on next page)

Box 1. *Continued*

REGRESSION

A class of machine-learning tasks where the desired output is from a continuous domain (as opposed to discrete values in classification tasks). For instance, the location of a cell in an image can be represented by two continuous values, the x and y coordinates of its center.

REGULARIZATION

A technique to inject expert knowledge into modeling. Through regularization, one can limit a learned model to a certain family. This is in contrast to letting the training algorithm be completely free in adapting to the training data. Regularization techniques often help with generalization of the model.

SUPERVISED, WEAKLY, SEMI-, SELF-, AND UNSUPERVISED LEARNING

Classical machine learning is divided into different categories based on the quality or quantity of annotation provided with the data. If the desired output of the model is provided for all of the training samples, it is called supervised learning. When this annotation is only provided for a subset of the training set, it is called semi-supervised learning. If the annotations are provided for all data samples but are not exactly what is required for the model to output, it is called weakly supervised learning. Unsupervised learning is when no annotation is available whatsoever. Self-supervised learning refers to the case that the desired output is in the sample itself and requires no annotation.

TEST SET

A held-out subset of the data that is used to measure the generalization performance of a final learned system to unseen examples.

TRAINING

The process of fitting the parameters of a model, often through an optimization procedure, such that a mapping function is learned between the input data and the desired output.

TRAINING SET

The set, comprising data samples and possibly their annotations, which is used to train the machine-learning model.

TRANSFER LEARNING

Transfer learning involves the process of transferring a model from one task with abundant training data to another (similar) task with less or no annotated training data.

VALIDATION SET

A held-out set that is used to tune the hyper-parameters of a machine-learning model as well as diagnosing its problems.

the mapping between the input samples and the labels. Most commonly used machine-learning algorithms fall into the supervised category. On the other hand, if no label is provided with the training samples, we call this "unsupervised learning." In this case, it is the algorithm's job to not only learn the mappings from the training examples to categories, but also to determine a reasonable set of categories from the training set. For instance, if we are not aware of the phenotype categories *a priori*, it is an unsupervised learning task to find those categories as well as to assign samples to them.

Different machine-learning models have different degrees of freedom (also known as "parameters") to learn the task at hand. Most of these parameters are automatically learned from training data using the algorithms provided with the model. However, some other parameters are left for the machine-learning experts to select for each task and training data individually. The latter are usually referred to as "hyper-parameters" of the model and the process of selecting them is called (hyper) "parameter tuning" (Bermudez-Chacon and Smith, 2015). Since parameter tuning is a manual process, it makes the training cumbersome and inefficient. Thus, models with less hyper-parameters are preferred.

When dealing with machine-learning models, there is a trade-off between providing field-expert knowledge to the model and letting the model to be completely free to learn everything from data. The former makes the learning process easier, but at the cost of being suboptimal, while the latter has the potential to be optimal but might be very hard to learn in practice and

Table 1. Links to Software

Software	Link to Code or Executable	References
CellProfiler	cellprofiler.org	Carpenter et al., 2006
BioConductor	bioconductor.org	Huber et al., 2015
Icy	icy.bioimageanalysis.org	De Chaumont et al., 2012
BioImageXD	bioimagexd.net	Kankaanpää et al., 2012
ImageJ/Fiji	fiji.sc	Schindelin et al., 2012
ADAPT plugin	bitbucket.org/djpbarry/adapt	Barry et al., 2015
NeuriteTracker	github.com/sgbasel/neuritetracker	Fusco et al., 2016
CPA	cellprofiler.org/cp-analyst	Jones et al., 2008
CellClassifier	pelkmanslab.org/?page_id=63	Rämö et al., 2009
Enhanced CellClassifier	provided as supplementary material with the author's permission	Misselwitz et al., 2010
Advanced CellClassifier	cellclassifier.org	Horvath et al., 2011
Phaedra	phaedra.io	Cornelissen et al., 2012
cellXpress	cellxpress.org	Laksameethanasan et al., 2013
HCS-Analyzer	hcs-analyzer.ip-korea.org	Ogier and Dorval, 2012
Ilastik	ilastik.org	Sommer et al., 2011
Trainable Weka Segmentation	imagej.net/Trainable_Weka_Segmentation	Arganda-Carreras et al., 2017
CecogAnalyzer	cellcognition.org	Held et al., 2010
CellCognition Explorer	software.cellcognition-project.org/explorer	Sommer et al., 2017
Cytomine	cytomine.coop	Marée et al., 2016a
WND-CHARM	github.com/wnd-charm/wnd-charm	Orlov et al., 2008
CP-CHARM	github.com/CellProfiler/CPCharm	Uhlmann et al., 2016
PhenoRipper	awlab.ucsf.edu/Web_Site/PhenoRipper	Rajaram et al., 2012
HTX	github.com/CarlosArteta/htx	Arteta et al., 2017
PhenoDissim	bioconductor.org	Zhang and Boutros, 2013
CellOrganizer	cellorganizer.org	Murphy, 2012

requires a lot of training data. An important instance of this question relates to how input samples are represented to a model. The samples' representation can be their raw sensory measurements (e.g., pixel values, in case of an image) or some higher-level information, designed by experts of the field and extracted from the measurements (e.g., indicators for size, shape, and texture of the cell). Many classic machine-learning methods require the hand-crafted representations to work well, while more recent and highly successful techniques including deep learning can automatically learn representation (although they require an abundance of training data).

Below, we describe the most popular software tools currently available that use machine learning to analyze cell-based assays. These tools vary substantially in terms of usability, functionality, interfaces, and performance. Links to the source code or executables of the software discussed in this review are provided in Table 1. We discuss these differences and provide a summary of the various features (Table 2).

Free and Open-Source Tools for Phenotypic Image Analysis

Over the past 10 years, a number of free and open-source software tools have been developed that are capable of performing phenotypic analysis on images of cell-based assays. One of the most well known is CellProfiler Analyst (CPA) (Jones et al., 2009). CPA features a graphical user interface (GUI) allowing the user to define a set of phenotypes and annotate individual cells accordingly (Figure 1A). Annotations created by the user can be used to train a supervised machine-learning algorithm. The user can then ask the software to "score" cells in a desired image or in multiple images. Scoring cells applies the trained machine-learning model to predict the phenotype of unseen cells.

The first version of CPA, published in 2008, marked an important milestone in phenotypic analysis (Jones et al., 2008). For the first time, biologists could easily apply modern statistical learning methods to recognize single-cell phenotypes automatically. It directly interfaced with the popular image analysis software CellProfiler, which provided cell segmentations and extracted measurements. The initial software release had several limitations: only two phenotypic classes (positive and negative) could be defined for classification, only one machine-learning algorithm, GentleBoost, was supported (Friedman et al., 2000), and the GUI made it difficult to explore the data effectively. Despite these limitations, the software saw widespread use. Recent updates have addressed many of these issues (Dao et al., 2016)—it is now possible to define multiple phenotypes, perform inference for single cells or whole images, and several machine-learning algorithms are supported (AdaBoost, Support Vector Machine [SVM], GradientBoost, Logistic Regression, Linear Discriminant Analysis, *k*-Nearest Neighbors, and GentleBoost). CPA now features an image viewer which allows the user to visualize scored cells within the image, and a plate heatmap viewer, which displays well-level readouts using a color-coded plate layout. CPA also allows the

Table 2. Cell Classification Software Comparison

	CellProfiler Analyst (v.2.2.1)	Cell Classifier	Enhanced Cell Classifer	Advanced Cell Classifier (v.3.0)	Phaedra (v.1.0.1)	cellXpress (v.1.4.2)	HCS-Analyzer (v.1.0.4.3)	Ilastik (v.1.2.2.4)	Trainable Weka Segmentation (v.3.2.13)	Cell Cognition (v.1.6.0)	Cell Cognition Explorer (v.1.0)	WND-CHARM (v.1.60)	Cytomine (v.1.0)	CP-CHARM	Pheno Ripper (v.2.0)
Documentation															
User guide	●	●	●	●	●	●	●	●	●	○	●	○	●	○	●
Website	●	●	○	●	●	●	●	●	●	●	●	●	●	●	●
Video tutorial	●	○	○	●	○	○	●	○	○	○	○	○	●	○	●
Open source code	●	●	●	●	○	○	●	●	●	●	●	●	●	●	●
Test dataset/demo	●	○	●	●	●	●	●	●	●	●	●	●	●	●	●
Usability															
No programming experience required	●	●	●	●	●	●	●	●	●	●	●	○	●	●	●
User-friendly GUI	●	●	●	●	○	○	●	●	●	○	●	○	○	○	●
Intuitive visualization settings	●	●	●	●	●	●	○	●	○	●	●	○	●	○	●
Does not require commercial licence	●	M	M	●	●	●	●	●	●	●	●	●	●	●	●
Portability on Win/ Linux/Mac.	●	●	●	●	Win/ Mac	Win/ Linux	Win	●	●	●	●	●	Linux	●	Win/ Mac
Functionality															
Plate/image selection	●	●	●	●	●	●	●	●	○	●	○	○	●	○	●
Time-lapse analysis	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○
3D analysis	○	○	○	○	●	○	○	●	●	○	○	○	○	○	○
Cell segmentation	○	○	○	○	○	●	○	●	●	●	●	○	●	○	○
Cell feature extraction	○	○	○	○	○	●	○	●	●	●	●	●	●	●	○

(*Continued on next page*)

Table 2. ***Continued***

	CellProfiler Analyst (v.2.2.1)	Cell Classifier	Enhanced Cell Classifer	Advanced Cell Classifier (v.3.0)	Phaedra (v.1.0.1)	cellXpress (v.1.4.2)	HCS-Analyzer (v.1.0.4.3)	Ilastik (v.1.2.2.4)	Trainable Weka Segmentation (v.3.2.13)	Cell Cognition (v.1.6.0)	Cell Cognition Explorer (v.1.0)	WND-CHARM (v.1.60)	Cytomine (v.1.0)	CP-CHARM	Pheno Ripper (v.2.0)
Supervised classification	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○
Automated phenotype discovery	○	○	○	●	○	○	○	○	○	○	●	○	○	○	●
Active learning	●	○	●	●	○	○	○	●	○	○	○	○	○	○	○
Similarity search	●	○	○	●	○	○	○	○	○	○	●	○	●	○	○
Output															
Visual cell classification	●	●	●	●	●	○	○	●	○	●	●	○	●	○	●
Feature-based statistics	○	○	○	●	●	●	●	○	○	●	○	○	●	○	○
Class-based statistics	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●
Plate-based statistics	●	●	●	●	●	●	●	○	○	○	●	○	○	○	○

●, available/yes; ○, not available/no; GUI, graphical user interface; M, MATLAB.

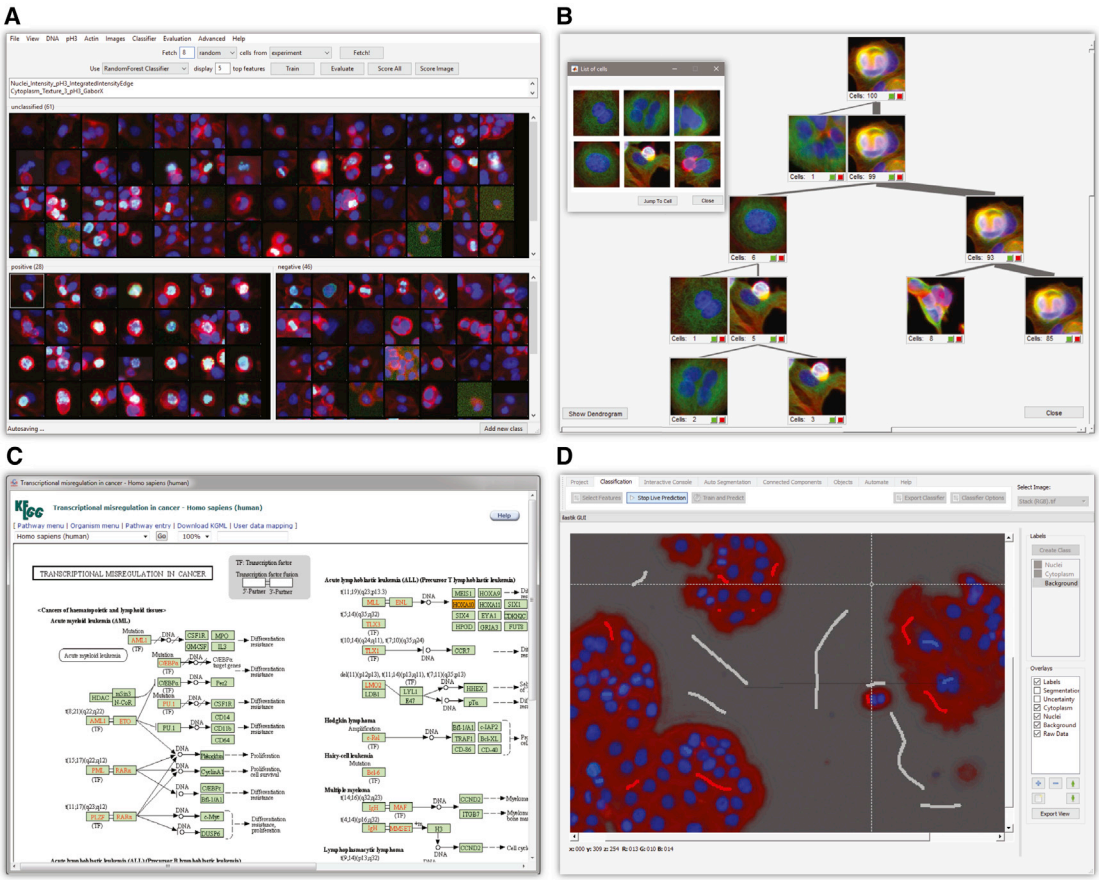

Figure 1. Free Software Tools for Phenotypic Image Analysis

This review covers phenotypic image analysis tools with a variety of capabilities (some overlapping). Here, we highlight key features of four software packages.

(A) A graphical user interface in CellProfiler Analyst allows users to fetch thumbnail images of cells from their data and drag-and-drop them into phenotypic classes. This method of annotation facilitates the training of a machine-learning algorithm, which then scores every cell in the dataset based on rules it learned from the training data.

(B) Finding novel phenotypes in a large dataset can be challenging. Advanced Cell Classifier v.2.0 helps users mine their data and identify novel phenotypes by organizing cells into a browsable tree where each node contains a group of cells with similar appearance. A new phenotypic class can be defined by selecting a cell or group of cells from the tree.

(C) HCS-Analyzer can display information about pathways involved in a certain phenotype. The software will parse the genes associated with the phenotype and gather the related pathways by interfacing with the Kyoto Encyclopedia of Genes and Genomes database.

(D) Ilastik features a simple, user-friendly process for interactively training a classifier to recognize objects or phenotypes. The user draws on the image to indicate a region belonging to a certain class or phenotype, this information is fed to a machine-learning algorithm which immediately displays the updated classifier predictions. The user can then identify and correct any mistakes made by the classifier.

user to explore their results using scatterplots, boxplots, density plots, and histograms. The GUI has been updated to include a gallery view, and it is now easier to fetch batches of cells for annotation (e.g., a trained classifier can retrieve cells from an image that it predicts are of a certain phenotype).

Following the release of CPA, several other software packages have appeared which also support machine-learning analysis applied to microscopic images of cell-based assays. CellClassifier (Rämö et al., 2009) and Enhanced CellClassifier (Misselwitz et al., 2010) were released shortly after CPA. Both packages offer similar functionalities. One important difference is the method used to find cells for annotation. In CPA, the software fetches randomly selected thumbnail images of cells and presents them for the user to annotate. While this may be a more efficient approach, it removes control from the user to choose which cells are most useful for annotation. Thumbnails of cells often fail to provide contextual information (e.g., appearance of its neighborhood). CellClassifier and Enhanced CellClassifier give the user more freedom to manually browse through images and choose which cells to annotate (although this approach may be more time consuming). CPA, CellClassifier, and Enhanced CellClassifier all rely on features extracted using external software such as CellProfiler for classification. CellClassifier and Enhanced CellClassifier implement one machine-learning algorithm, SVM, but they do support multi-class classification (i.e., they can classify more than two phenotypes).

An important shortcoming common to these tools is that they assume the user has prior knowledge of the interesting phenotypes present in the data, or they can easily be found by manual inspection. This can be a dangerous assumption, as rare but important phenotypic classes can potentially be overlooked. Investigators often collect large image datasets for original research that are the first (and only) of their kind. In many cases,

it is not clear beforehand which phenotypes are interesting or how they will appear. Even after the data are collected, the scale of the data often makes it impractical or impossible to identify all the important phenotypes by inspection.

To address this problem, we released an update to the phenotypic image analysis software package Advanced Cell Classifier (ACC) (Horvath et al., 2011) in 2017 with data exploration and phenotype discovery tools (Piccinini et al., 2017). ACC includes a phenotype finder tool that helps users identify novel phenotypes in their data by organizing the cells into a browsable hierarchy that groups cells by their appearance (Figure 1B). It also features example-based mining of the data—more examples of cells of a rare phenotype can be found quickly by providing the tool with a query example. The software features active learning for annotation, which makes more efficient use of the expert's time by prioritizing the most informative examples and presenting them for annotation. Active learning helps train an accurate classifier more efficiently and avoids redundant annotations (Smith and Horvath, 2014). In addition to these discovery tools, ACC shares many similar functionalities to CPA such as an interactive GUI to view images, support for 16 machine-learning algorithms, data visualizations, and an interface with CellProfiler.

Phaedra (Cornelissen et al., 2012) and cellXpress (Laksameethanasan et al., 2013) are open-source platforms for visualization and analysis of HCS data. They feature intuitive and user-friendly GUIs allowing similar data visualizations as CPA, including plate heatmaps, charts, tables, image viewers, and dose-response curves. In contrast to previous tools, cellXpress does not require external software such as CellProfiler to perform image analysis, it is a self-contained solution. Phaedra allows image analysis data to be imported from Columbus commercial software or a MATLAB interface to define your own custom image analysis (importing CellProfiler image analysis data is a planned feature). Neither Phaedra or cellXpress have native support for automatic phenotype classification, nor do they provide a GUI for the user to directly annotate images for training. cellXpress allows cell and feature data to be exported, which can be loaded in *R*, a scripting language for statistical computing. There is no GUI for annotating individual cells, so phenotypes must be identified at the well level. The user must be proficient with the R scripting language and machine learning to perform the analysis, although some instructions are provided in the cellXpress documentation. cellXpress has an advantage over other software in processing speed. Most other software packages are written in relatively slow scripting languages such as Python or MATLAB, while cellXpress is written in C++, which is compiled to fast machine code and uses dynamic job scheduling and parallel processing to make full use of multi-core central processing units. Phaedra supports single-cell annotations by selecting groups of cells in an image, table, or plot, but only for manual classification. For data mining, Phaedra uses an interface to the Konstanz Information Miner (KNIME) graphical framework (Berthold et al., 2009). While KNIME is a very flexible GUI, a user must define a custom workflow in order to perform phenotype classification, which may be challenging for non-experts. An important feature offered by Phaedra is support for JPEG2000 (Taubman and Marcellin, 2012) and HDF5. This solves an important issue ignored by most other software packages—efficient image compression for quick network access and efficient storage.

HCS-Analyzer (Ogier and Dorval, 2012), like Phaedra and cellXpress, is a software solution designed for high-content screening. HCS-Analyzer works with features extracted by another image-processing software such as CellProfiler or Columbus. It does not work with the original images, so it is not possible to visualize the cells or to annotate them by their appearance. Data can be imported through a comma-separated value file, and the GUI allows the user to visualize the extracted features in a number of ways including plate heatmaps, histograms, scatterplots, dose-response curves, and hierarchical tree maps. Phenotype classification is performed either in a supervised (SVM, neural network, *k*-nearest neighbors, or random forest) or unsupervised manner (*k*-means, expectation maximization, or hierarchical clustering). Unsupervised phenotype classification requires the user to provide the number of classes and features that should be considered, and the algorithm groups cells without any labeling. HCS-Analyzer features an error identification and correction system that automatically identifies and corrects systematic errors, such as edge or dispensing effects. It does this by clustering features and testing if the plate geometry explains the feature clusters. HCS-Analyzer software interfaces with the Kyoto Encyclopedia of Genes and Genomes database (Kanehisa and Goto, 2000), and is able to compute and visualize gene and pathway information relevant to hits from the screen data (Figure 1C).

Ilastik is a simple, user-friendly tool for segmentation, classification, and analysis (Sommer et al., 2011). At the core of Ilastik is an innovative interactive training process that allows the user to iteratively train and correct the machine-learning model using online semi-supervised learning. The user annotates by drawing on the image with a brush tool to indicate regions belonging to a certain class or phenotype, and this information is fed to a machine-learning algorithm, which immediately displays the updated classifier predictions (Figure 1D). This approach allows the user to easily identify and correct the classifier's mistakes. Ilastik's approach gives it great versatility—it can be used for pixel classification, object classification, tracking, counting, and segmentation in images (Palazzolo et al., 2012) and volumes (Nunez-Iglesias et al., 2013). To work in real-time, Ilastik must preprocess image features and retain them in memory, which limits Ilastik to work with only a small set of images at once (typically less than ten). If the user is satisfied with prediction results, a batch mode can be used to apply the trained model on larger amounts of data. However, if interesting biological phenomena appear outside of the image set retained in memory for interactive learning, they will likely be missed. Ilastik supports HDF5 files, import/export from Fiji, and can export segmentations to CellProfiler. Trainable Weka Segmentation (TWS) (Arganda-Carreras et al., 2017) mimics the interactive training process from Ilastik and integrates it as a plugin for the popular software Fiji/ImageJ. Like Ilastik, the interface of TWS allows the user to iteratively train and correct the classifier, and supports 2D and 3D image data. TWS offers some improvements over Ilastik. While Ilastik uses a random forest classifier, and relatively few image features, TWS supports all the machine-learning algorithms supported by Weka (Frank et al., 2016) and over 20 features for 2D images. It is also possible to employ user-designed

image features or classifiers in TWS. Although it is designed for segmentation, pixel-based classification in TWS can be used to recognize phenotypes.

CellCognition (Held et al., 2010) is a computational framework capable of performing phenotype recognition in time-lapse data. CecogAnalyzer is an application built on top of the CellCognition framework that detects and segments cells in fluorescence images, extracts features, tracks the cells over time, and classifies cellular phenotypes. A notable feature of CecogAnalyzer is the ability to reliably recognize time-dependent phenotypes, such as stages of mitosis. A GUI allows the user to visualize the data and create annotations for training. The machine-learning models behind CecogAnalyzer includes an SVM that predicts the phenotypic classes at individual times steps, and a Hidden Markov Model (HMM) that corrects noise in the predictions and enforces smoothness over time. CecogAnalyzer is designed to be used on time-lapse data, but it is possible to apply it to static images. It lacks some of the data visualizations offered by other software such as the plate heatmap. CecogAnalyzer supports multi-well plates and batch processing, which allows large experiments to be processed in a cluster environment.

CellCognition Explorer is an all-in-one software tool for image preprocessing, image segmentation, feature extraction, and supervised phenotype classification (Sommer et al., 2017). CellCognition Explorer is developed by the same team as CecogAnalyzer, but is not intended for temporal data. Like the phenotype finder and similar cell search tools in Advance Cell Classifier v.2.0, CellCognition Explorer aims to help the user to quickly discover abnormal cell morphologies within large datasets through powerful tools to mine datasets for rare phenotypes. It uses novelty detection algorithms to autonomously learn intrinsic cell-to-cell variability in an untreated negative control population, which sensitizes a classifier toward perturbation-induced phenotypes. Abnormal cells are then scored based on weighted distance to the mean of a control population (using a choice of metrics). CellCognition Explorer is designed to work with conventional hand-designed cell-based measurements of intensity, shape, texture, granularity, etc., or, it can work with unsupervised features learned using auto-encoders through a separate module. CellCognition Explorer includes a responsive cell gallery GUI that allows cells to be interactively sorted according to their similarity, treatment, or classification.

Cytomine is an internet application that allows remote visualization, collaborative annotation, and (semi-)automated analysis of very large high-resolution images (Marée et al., 2016a). It uses modern web technologies, databases, and machine learning to organize, explore, share, and analyze multi-gigapixel imaging data. Cytomine was originally developed to analyze bright-field cytology and histology images, but it has been applied to various other types of imaging datasets, including fluorescent cell images. Cytomine supports image segmentation, object retrieval, interest point detection, and object sorting. Users can upload an image to a project and collaboratively annotate the image. Every time a user annotates a new object in the image, an unsupervised, incremental, content-based image retrieval method displays similar annotations in the database. Finally, a scale invariant recognition model can be applied in order to analyze the content of the images at different resolutions and make predictions for novel annotations (Marée et al., 2016b).

WND-CHARM is a software tool that takes a holistic approach to phenotype analysis. It is designed to remove the need for parameter tuning (Orlov et al., 2008). It not only relies on single-cell segmentation for analysis, but considers the image as a whole. WND-CHARM computes a number of features on the entire image, including polynomial decompositions, high-contrast features, pixel statistics, and texture features. It selects the most informative features according to Fisher's discriminant score, which finds features that maximize distance between classes and minimize distance within classes. It then performs classification using weighted-neighbor distances on the selected features. For training, WND-CHARM only requires image-level annotations, and therefore does not provide a GUI for annotating. For simple assays, the holistic parameter-free approach of WND-CHARM is appealing, but it may struggle to correctly identify images with subtle phenotypes or heterogeneous populations. WND-CHARM is a command-line tool, which requires some technical expertise to operate. Recently, a new implementation of WND-CHARM with a GUI was released to interface with CellProfiler called CP-CHARM (Uhlmann et al., 2016). However, it requires external modules to execute and is not supported by all releases of CellProfiler (support for versions 2.0.11710 and version 2.1.0).

PhenoRipper, like WND-CHARM, uses a segmentation-free approach to phenotypic profiling in an effort to make it fast and simple (Rajaram et al., 2012). PhenoRipper breaks the image down into a square grid of blocks, and performs analysis on the blocks rather than cells. As such, it does not quantify some important properties of the cell such as area or average intensity. The user is asked to provide only a few simple parameters: an intensity threshold to segment cells, and the number of pixels that make up a block. PhenoRipper uses the intensity threshold to separate cells from the background. Each cell is composed of multiple blocks, and PhenoRipper uses an unsupervised approach to group blocks with similar appearance into a fixed number of clusters. Analysis is performed on the classified blocks within each image. The results can be explored through several different visualizations provided with the software, including a clustergram to visualize how block types correlate with experimental conditions.

Perspectives and Challenges

Over the past decade, we have witnessed many changes to the way phenotypic image analysis is performed, as demonstrated by the diversity of approaches in the software listed above. Recent advances in screening technologies and artificial intelligence suggest dramatic and exciting shifts in the near future. But, while we look ahead, we must also acknowledge the limitations of the current technologies, and the fact that some of the challenges we face today will become more pronounced as we demand more from phenotypic image analysis. In this section, we provide an outlook on what we perceive as future directions and challenges in phenotypic image analysis.

Big Data and Phenotypic Image Analysis

Our capacity to generate and store image data far outstrips our ability to analyze it manually. This fact is the impetus for phenotypic image analysis. But even using software to automate

phenotypic image analysis, there is no guarantee that our ability to transfer and process image data is sufficient to handle the rate at which we can now generate it. Indeed, there are many cases where images can be acquired faster than they can be processed and stored. Pushing the limit of modern high-throughput microscopes, it is possible acquire 5–10 images per second in simplified assays. At this rate, an entire 384-well plate can be imaged in 3–4 min. A high-throughput laboratory running just a single microscope has the potential to generate over 10 TB of raw image data on a daily basis (the typical camera resolution of an HCS microscope varies between 1 and 4 megapixels, with each pixel encoded in 8, 12, or 16 bits, and the typical file size being 1–8 MB for an image of a single field). The time required to perform basic image processing using current software can be orders of magnitude slower than image acquisition. Image processing, segmentation, and feature extraction can take up to several minutes per image.

Traditionally, data have been stored and processed in-house, but the throughput of modern microscopes creates computational demands too great for modest laboratory resources. To obtain results in a reasonable time, high-performance computing (HPC) is required. However, moving the data to an HPC cluster or cloud computing infrastructure introduces a new bottleneck—transfer speed. The challenge is: how can we adjust our data pipelines so that processing keeps pace with the rate we generate data? One potential solution is improved compression. Solutions such as JPEG2000 (Taubman and Marcellin, 2012) and HDF5 (Dougherty et al., 2009) can reduce the volume of complex image data by orders of magnitude. Another solution is to perform real-time image preprocessing at the microscope. This approach can reduce the data throughput by extracting essential features and avoid working with raw images altogether.

Understanding the Data: Quality and Completeness

Annotation, the process of creating an expertly labeled dataset for supervised learning is arguably one of the most important steps in a phenotypic screen. However, worryingly little attention is given to this crucial step of the analysis. As a result, datasets used for phenotypic image analysis are frequently too small, uninformative, or incomplete. In many of the software solutions described in the previous sections, annotation collection and training the algorithm are done simultaneously, in what is known as online learning. A few examples are annotated, the classifier is updated, then more annotations are collected, etc. Typically, only a few hundred to a few thousand examples are collected in total. There is often little or no feedback to check if enough annotations have been collected for training to converge or for the classifier to generalize. Some software tools provide cross-validation accuracy or a confusion matrix when the model is trained, but this is the extent of annotating/training feedback.

It is often unknown which phenotypes will be present in an experiment or how they will appear. If a phenotype is rare or the dataset is large, phenotypes are likely to be overlooked. Some methods for data mining and discovery exist that can help ensure completeness of training data (Figure 2A). ACC provides a tool to help uncover rare phenotypic classes. CellCognition Explorer features a novel phenotype detection framework. HTX provides a method to concisely visualize cell similarity over an entire screen (Arteta et al., 2017). A method for computing phenotypic dissimilarity between cell populations, PhenoDissim, is available as an R package (Zhang and Boutros, 2013). Another common issue with annotations is class imbalance—certain phenotypes may occur less frequently causing imbalanced datasets, which impede classification performance. In practice, it may be difficult to find sufficient examples of a particular class. To address this, ACC and CellCognition Explorer can find cells with similar appearance when provided with an example. CPA makes it possible to use a trained classifier to find examples predicted to be of a certain class.

The interface used for annotation directly affects quality and quantity of the annotations, and consequently affects the performance of the classifier. Unfortunately, many existing tools provide inadequate functionalities, or exclude annotation completely. A good GUI should be responsive, intuitive, efficient, and provide useful feedback as to how the training is progressing. Some GUIs work with thumbnail images of cells. While this can increase efficiency, it can also be important to view the cell in the context of the full image. Some interfaces, such as ACC, use active learning to suggest examples that the classifier is least certain about in order to avoid wasting effort on uninformative examples.

Few phenotypic image analysis tools make use of unsupervised, semi-supervised, or weakly supervised learning. In light of the high cost of collecting expert annotated data, this seems to be a missed opportunity. Semi-supervised systems take unlabeled data into account as well as annotated examples. Weakly supervised (or bootstrapping) methods use a form of self-training on even fewer annotated examples. Among the software listed above, HCS-Analyzer and PhenoRipper are the only packages to support unsupervised learning out of the box.

Continuous Modeling of Biological Processes

Many interesting biological processes are continuous by nature. For example, cell differentiation (Trapnell et al., 2014), cell development (Bendall et al., 2014), cell adhesion, and cell death. Many other biological processes have important continuous aspects, such as endocytosis or drug uptake. Most existing phenotypic analysis tools rely on machine-learning algorithms designed for classification, even though discrete categorization does not reflect the biological reality. To make use of these tools, researchers are faced with the difficult task of defining artificial cutoffs to discretize inherently continuous processes. For example, the continuous process of infection of a cell by a pathogen might be arbitrarily broken into ‘‘uninfected,’’ ‘‘weakly infected,’’ and ‘‘strongly infected.’’ CellCognition was one of the first attempts to apply machine learning to recognize patterns in a continuous biological process (Held et al., 2010). In a pioneering effort to model the cell cycle, it predicted each cell to be in one of seven mitotic states at every step in a time series, and it enforces probable transitions between these states with an HMM. However, the underlying modeling in CellCognition is not truly continuous—the cell cycle is broken into discrete phases by the classifier. Another recent work using deep learning to classify cell phases from raw images suffers from the same problem (Eulenberg et al., 2017).

Regression algorithms, in contrast to classification methods, are designed to make continuous-valued predictions. In an effort to characterize the process of influenza A entry to human cells,

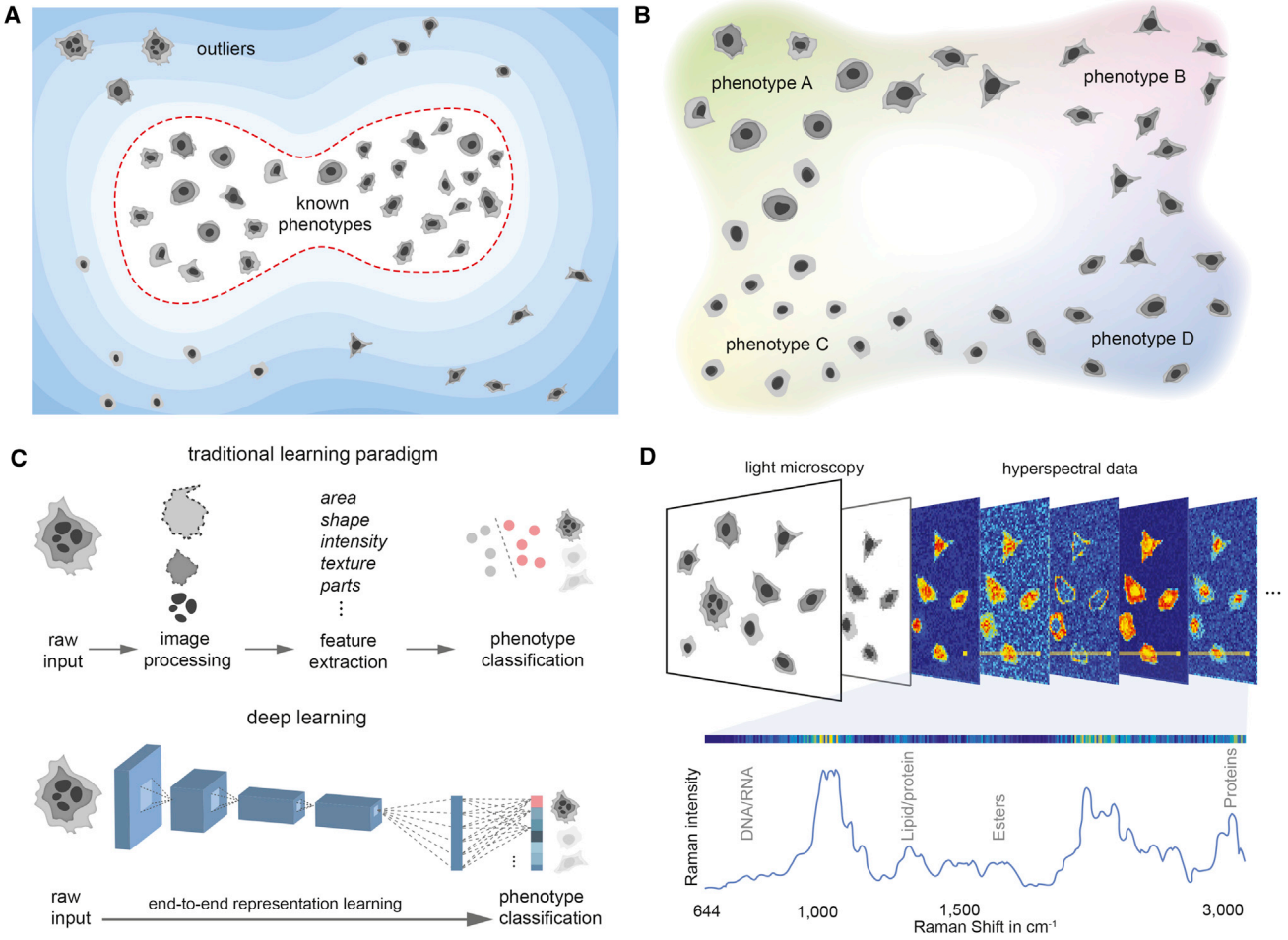

Figure 2. Trends in Phenotypic Image Analysis

In this figure, we highlight some of the most promising developments in phenotypic image analysis (according to the authors).

(A) Novelty detection methods can help identify rare phenotypes buried within large datasets. Given a set of annotations of known phenotypes, these methods can predict which cells are most dissimilar to the known population and present them to the user. The outliers are likely to be novel phenotypes.

(B) Many biological processes are continuous by nature and cannot be discretely categorized. One way to train a machine-learning algorithm to recognize continuous processes such as cell differentiation is to sort cells by appearance in a 2D plane. The locations in the embedded space can be used to train regression algorithms to make continuous-valued predictions between phenotypic extremes.

(C) In the traditional paradigm to learning, dedicated algorithms process the raw input data, extract hand-designed features, and then provide the features to a machine-learning algorithm, which is designed to perform classification. In contrast, deep learning is an end-to-end approach to learning that takes raw images as input and produces the desired output. The internal representations necessary for classification are learned automatically instead of being designed by hand and provide unprecedented discriminatory power.

(D) New hyperspectral imaging technologies such as imaging mass spectrometry and Raman spectrometry, which perform molecular profiling, can be multiplexed with traditional light microscopy. This rich combination of information describing the phenotype and genotype can provide new insights that neither modality could accomplish alone.

Yamauchi et al. (2011) and Banerjee et al. (2014) used a regression algorithm to predict the ratio of scattered to packed endosomes within the cell. Software designed for these studies tasked experts to sort cells according to the phenotype using a GUI. Users arranged thumbnails of cells with one extreme phenotype on one side of a continuum (packed endosomes), and the other extreme phenotype on the other (scattered endosomes). The machine-learning algorithm used this arrangement of the training data to predict the scattering index of new cells. While the regression approach in this work modeled the continuous process of infection with better fidelity, it had important shortcomings. Some biological processes are difficult to model in a 1D space, such as mechanisms with conditional dependencies, cycles, or branching. One solution is to model biological processes in higher-dimensional spaces, such as a 2D regression plane—an embedded space where phenotypes are sorted by appearance (Figure 2B). By arranging training data in this visually intuitive manner, the user can train a machine-learning algorithm to model phenotypic extremes and the continuous biological processes between these extremes. This capability is implemented in the latest version of ACC.

We have only taken the first steps toward an accurate and intuitive solution to model biological processes in a continuous way. There is a great need to improve our models of continuous processes, and efforts will likely continue and better tools will be developed in the coming years.

Deep Learning

Deep learning is an end-to-end approach to learning. It takes raw data (images in the case of phenotypic profiling) as input and produces the desired output by learning from training examples. This eliminates the need for hand-crafted features, which are cumbersome to design and give suboptimal discriminatory power. In deep learning, a network of artificial neurons organized into layers embeds all the necessary processing. Each layer takes the output of the previous layer as input, and applies simple operations that transform the previous representation into a higher, slightly more abstract representation. Although the individual computations are simple, unprecedented discriminatory power is achieved through the compositional nature of the architecture. Modern networks may contain hundreds of layers (He et al., 2016). In contrast to the traditional approach (image processing, feature extraction, and classification), the end-to-end approach of deep learning is attractive because it offers a holistic solution and often yields better performance (Figure 2C).

Deep networks have been successfully applied to different visual recognition tasks since their ground-breaking performance in the 2012 ImageNet challenge (Krizhevsky et al., 2012) including pose estimation, object tracking, object retrieval, activity recognition, super-resolution, etc. The state-of-the-art for several important tasks used in phenotypic image analysis is now dominated by deep learning approaches, including semantic segmentation, feature extraction, image enhancement, and object recognition.

One of the first works to use deep learning for phenotypic image analysis was U-Net (Ronneberger et al., 2015). U-Net is a novel deep network architecture designed to learn segmentation. It achieved first place in the cell-tracking challenge of the IEEE International Symposium on Biomedical Imaging 2015. However, it does not address phenotypic cell classification. One of the first studies to apply deep learning for phenotype recognition was (Dürr and Sick, 2016). They applied a deep convolutional network on 40,000 images from the Broad Bioimage Benchmark Collection (BBBC022v14, the "Cell Painting" assay [Gustafsdottir et al., 2013]). They show superiority of the end-to-end deep learning approach compared with a traditional pipeline using hand-designed CellProfiler features and machine-learning methods such as SVM, random forests, and Fisher linear discriminant. A similar study by Pawlowski et al. (2016) showed that off-the-shelf features from networks trained on conventional images outperform the traditional paradigm.

Deep networks have a number of advantages over the traditional paradigm. One advantage is that deep networks can learn to do multiple tasks using the same network. For example, Kraus et al. (2016), perform segmentation and phenotypic classification in a unified network. Due to its ability to model highly complex functions, deep learning can tackle problems that are infeasible with traditional approaches. Kraus et al. (2017) used deep networks for subcellular protein localization and show that, unlike traditional approaches, the models can be successfully transferred to datasets with different genetic backgrounds acquired from other laboratories, even with abnormal cellular morphology. Another important aspect of deep learning is that the features learned are highly general and thus transferable (Azizpour et al., 2016), even to unseen tasks and images. Pärnamaa and Parts (2017) trained a deep network on subcellular localization of certain proteins, and transferred the learned features to unseen proteins and cellular compartments where only a few images were available. Winsnes et al. (2016) also address protein subcellular localization for HCS using deep networks with multi-label classification. Deep networks can be taught to imitate data or the output of another algorithm by providing data samples and the corresponding results of the algorithm using generative adversarial networks (GANs) (Goodfellow et al., 2014). Sadanandan et al. (2017a) use this approach to segment bright-field images of cells, while Arbelle and Raviv (2017) use this approach to segment cells in fluorescence images. Deep networks have recently been employed to perform other advanced phenotypic analyses, such as novelty detection (Sommer et al., 2017; Sailem et al., 2017), phenotypic changes due to malignancy (Wieslander and Forslid, 2017), classification of label-free cells in phase contrast imaging (Chen et al., 2016), and spheroid segmentation in various microscopy conditions (Sadanandan et al., 2017b). Deep networks have also been used to improve the quality of microscopy images, by increasing resolution beyond the diffraction limit (Zhang et al., 2018) and by restoring defects in the image (Weigert et al., 2017).

The majority of these deep learning works use open-source deep learning frameworks such as Caffe, TensorFlow, and PyTorch. As a result, the code is more easily shared, adopted, and improved upon. Many of these deep learning works publicize their code (Ronneberger et al., 2015; Kraus et al., 2016; Pärnamaa and Parts, 2017; Sommer et al., 2017) and are integrated with existing phenotypic analysis software.

Considerable progress has been made by deep learning in phenotypic analysis of cell images, and we can expect accelerated progress in the future. Deep networks flourish with large amounts of training data—but, so far, the datasets used for training deep networks in bioimaging have been mostly small scale (Ching et al., 2018). Large biomedical datasets are starting to emerge; such as Medical ImageNet (Medical Image Net, 2018) and BBBC (Ljosa et al., 2012). Features trained on these large datasets can be transferred to similar tasks with less training data, replacing hand-crafted features. In the same manner, expert-designed pipelines used for image preprocessing and filtering may be replaced by deep networks. We can also expect novel problems to be addressed with deep learning, which are currently unexplored due to their complex nature.

Deep learning has many advantages, but it comes with its own challenges. One is that deep learning in tasks other than fully supervised learning has not been particularly successful—this includes unsupervised, semi-supervised, and active learning. Furthermore, tuning the hyper-parameters of a deep network is time consuming and requires substantial practical experience. It can be costly to train a deep network from scratch on a new task. Moreover, due to the complex structure of deep networks, interpreting its decisions is harder than linear models.

Interpretability of Machine Learning

While sophisticated machine-learning algorithms can outperform humans at recognizing phenotypes (Horvath et al., 2011), the models they employ can be difficult to interpret. This is problematic because our goal is often not just to recognize phenotypes with the greatest accuracy, but also to gain an understanding of what factors are most predictive of a phenotype. This understandability can lead to crucial biological insights.

Unfortunately, the reasoning behind the predictions of many of the most powerful classification algorithms cannot be easily explained. Predictions from SVMs with complex kernels, random forests, boosting, and neural network-based approaches (including deep learning) are often impossible to explain. Their decision is based on a highly complex non-linear combination of the input feature values. Some classification models are more interpretable. Decision trees, for example, can be readily interpreted as a series of binary decisions (e.g., is the mean cell intensity greater than a certain value? If so, is a certain texture present above a threshold value?). GentleBoost is the default classification method in CPA, which relies on a linear combination of a series of simple rules (e.g., intensity higher than a threshold). So long as the number of rules remains small, the explanation can be more-or-less understood.

As a general rule, more interpretable models are outperformed by their "black box" counterparts. But there are methods to demystify these models. Methods such as sequential feature selection (Aha and Bankert, 1996) can provide insight on how important a particular measurement is for classification, and local interpretable model-agnostic explanations can highlight regions of the image that are more sensitive for classification (Ribeiro et al., 2016). The interpretability problem is most apparent in deep learning, where many have criticized the black box nature of such complex networks (Castelvecchi, 2016). Various ways have been proposed to detect the regions of the input signal that are most responsible for the final prediction of a deep network (Oquab et al., 2015). However, since the most successful networks have on the order of one hundred layers, it is prohibitively hard to explain *how* those regions contributed in the final prediction.

Generative Modeling

Computational models of the a cell, including the cell phenotype, can be distinguished as either discriminative or generative. Discriminative models which learn to recognize the state of a cell based on observations such as its appearance, or generative models, which can synthesize new examples of cells in a particular state. The software reviewed above, and indeed most profiling methods, use a discriminative approach. Generative approaches capture variation in a population and encode it as a probability distribution, or generative models, which can use this information to synthesize new examples of cells in a particular state. The CellOrganizer software package is able to generate models of individual cells by modeling the structure of subcellular compartments on data from high-resolution microscopy images (Murphy, 2012). CytoGAN is a recent approach that trains GANs to synthesize realistic cell images that are useful for exploring morphological variations within or between populations of cells (Goldsborough et al., 2017). A similar work, using data from the Allen Cell Explorer, trained conditional auto-encoders to learn variation in cell structure and organization in order to synthesize high-quality single-cell images of a desired phenotype (Johnson et al., 2017).

Correlative Microscopy

In this article, we have mainly focused on how computational factors may impact phenotypic analysis under the assumption that the data sources will remain more-or-less as they are today. However, promising lines of research suggest that conventional light microscopy data may soon be combined with molecular profiling (e.g., mass spectrometry or Raman spectroscopy) or other imaging modalities, such as super-resolution or electron microscopy. These approaches promise great potential for discovering relevant spatial molecular information and building genotype-phenotype maps (Masyuko et al., 2013). Hybrid correlative techniques have gained recent attention for their ability to make the best of both worlds. These techniques combine spectral or molecular information with optical data. We foresee that, in the future, these combined data will provide new insights that neither modality could alone (Figure 2D).

Imaging mass spectrometry (Bodzon-Kulakowska and Suder, 2016; Palmer et al., 2017) is a technique designed to systematically measure the spatial molecular distribution of biomarkers, metabolites, peptides, or proteins by their molecular masses. It generates a hyperspectral image in which each pixel contains a mass spectrum with dimensions on the order of 10,000. However, the resolution is limited (typically 1–5 μm). One important computational task is to correlate the molecular profiling data with a light microscopy image, and to intelligently interpret the multiplexed data. This technique has gained increasing popularity recently (Liu et al., 2017), Another approach is Raman spectroscopy, which provides a structural fingerprint that can be used to identify molecules. It provides very high-dimensional spectral data, usually tens of thousands of dimensions, and its spatial resolution is in the range of optical microscopy limits (Smith and Dent, 2005). It is capable of single-cell analysis as well (Wagner, 2009; Schie and Huser, 2013). Finally, correlative light electron microscopy is a well-established technology for highly detailed spatial analysis of objects identified by light microscopy. The spatial resolution of electron microscopy can be few thousand times higher than light microscopy (Razi and Tooze, 2009).

Summary

In the past decades, the push toward systems biology has encouraged a holistic approach to understanding interactions between biological systems. As a result, biology has become much more interdisciplinary as fields such as genomics, proteomics, and bioinformatics become increasingly important. New technologies and tools are constantly being developed, many of which require highly specialized knowledge. One of the most important changes is our reliance on computational methods. Methods such as phenotypic image analysis are used routinely in research and in the pharma industry, where the search for new drugs or fundamental biological knowledge requires software analysis of images on a massive scale. These tools are becoming increasingly more powerful, but knowledge and training are required to properly apply these methods. More biologists are becoming competent users of these software tools (and even developers). As biologists become more computationally literate, they are increasingly able to solve problems themselves and develop a more complete understanding of the data. As a result, the quality of the science improves. By the same token, as computer scientists improve their knowledge of biology, they will be better able to recognize opportunities to apply their skill sets. The benefits of fostering interdisciplinary thinking are clear, and we should invest resources into broadening the knowledge of developers and experimentalists, so that the line between the two starts to disappear.

ACKNOWLEDGMENTS

The authors wish to thank Benjamin Misselwitz for providing source code of Enhanced CellClassifier. P.H. acknowledges support from the Finnish TEKES FiDiPro Fellow Grant 40294/13. F.P. acknowledges support from the European Association for Cancer Research (EACR) for a granted travel fellowship (ref. 573) and from NEUBIAS COST Action (European Cooperation in Science and Technology) for a granted short-term scientific mission (ref. CA15124). B.T., K.K., T.D., and P.H. acknowledge support from the European Regional Development Funds (GINOP-2.3.2-15-2016-00006 and GINOP-2.3.2-15-2016-00026).

AUTHOR CONTRIBUTIONS

K.S., F.P., T.B., K.K., T.D., H.A., and P.H. contributed to writing the manuscript and approved the final manuscript.

REFERENCES

Aha, D.W., and Bankert, R.L. (1996). A comparative evaluation of sequential feature selection algorithms. In Learning from Data, D. Fisher and H.-J. Lenz, eds. (Springer), pp. 199–206.

Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K.W., Schindelin, J., Cardona, A., and Sebastian Seung, H. (2017). Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. Bioinformatics *33*, 2424–2426.

Arbelle, A., and Raviv, T.R. (2017). Microscopy cell segmentation via adversarial neural networks. arXiv, 1709.05860.

Arteta, C., Lempitsky, V., Zak, J., Lu, X., Noble, A., and Zisserman, A. (2017). HTX: a tool for the exploration and visualization of high-throughput image assays. bioRxiv. https://doi.org/10.1101/204016.

Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., and Carlsson, S. (2016). Factors of transferability for a generic convnet representation. IEEE Trans. Pattern Anal. Mach. Intell. *38*, 1790–1802.

Banerjee, I., Miyake, Y., Nobs, S.P., Schneider, C., Horvath, P., Kopf, M., Matthias, P., Helenius, A., and Yamauchi, Y. (2014). Influenza A virus uses the aggresome processing machinery for host cell entry. Science *346*, 473–477.

Barry, D.J., Durkin, C.H., Abella, J.V., and Way, M. (2015). Open source software for quantification of cell migration, protrusions, and fluorescence intensities. J. Cell Biol. *209*, 163–180.

Bendall, S.C., Davis, K.L., Amir, E.A.D., Tadmor, M.D., Simonds, E.F., Chen, T.J., Shenfeld, D.K., Nolan, G.P., and Pe'er, D. (2014). Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. Cell *157*, 714–725.

Bermudez-Chacon, R., and Smith, K. Automatic problem-specific hyperparameter optimization and model selection for supervised machine learning. Technical Report, ETH Zurich, 2015.

Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., and Wiswedel, B. (2009). KNIME-the Konstanz information miner: version 2.0 and beyond. ACM SIGKDD Explorations Newsletter *11*, 26–31.

Bickle, M. (2010). The beautiful cell: high-content screening in drug discovery. Anal. Bioanal. Chem. *398*, 219–226.

Bodzon-Kulakowska, A., and Suder, P. (2016). Imaging mass spectrometry: instrumentation, applications, and combination with other visualization techniques. Mass Spectrom. Rev. *35*, 147–169.

Boutros, M., Heigwer, F., and Laufer, C. (2015). Microscopy-based high-content screening. Cell *163*, 1314–1325.

Brenner, S. (1974). The genetics of *Caenorhabditis elegans*. Genetics *77*, 71–94.

Caicedo, J.C., Cooper, S., Heigwer, F., Warchal, S., Qiu, P., Molnar, C., Vasilevich, A.S., Barry, J.D., Bansal, H.S., Kraus, O., et al. (2017). Data-analysis strategies for image-based cell profiling. Nat. Methods *14*, 849–863.

Carpenter, A.E., Jones, T.R., Lamprecht, M.R., Clarke, C., Kang, I.H., Friman, O., Guertin, D.D., Chang, J.H., Lindquist, R.A., Moffat, J., et al. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. *7*, R100.

Castelvecchi, D. (2016). Can we open the black box of AI? Nat. News *538*, 20.

Chen, C.L., Mahjoubfar, A., Tai, L.C., Blaby, I.K., Huang, A., Niazi, K.R., and Jalali, B. (2016). Deep learning in label-free cell classification. Sci. Rep. *6*, 21471.

Chia, N.Y., Chan, Y.S., Feng, B., Lu, X., Orlov, Y.L., Moreau, D., Kumar, P., Yang, L., Jiang, J., Lau, M.S., et al. (2010). A genome-wide RNAi screen reveals determinants of human embryonic stem cell identity. Nature *468*, 316.

Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., Kalinin, A.A., Do, B.T., Way, G.P., Ferrero, E., Agapow, P.M., Zietz, M., Hoffman, M.M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. bioRxiv. https://doi.org/10.1101/142760.

Cornelissen, F., Cik, M., and Gustin, E. (2012). Phaedra, a protocol-driven system for analysis and validation of high-content imaging and flow cytometry. J. Biomol. Screen. *17*, 496–506.

Dao, D., Fraser, A.N., Hung, J., Ljosa, V., Singh, S., and Carpenter, A.E. (2016). CellProfiler analyst: interactive data exploration, analysis and classification of large biological image sets. Bioinformatics *32*, 3210–3212.

De Chaumont, F., Dallongeville, S., Chenouard, N., Hervé, N., Pop, S., Provoost, T., Meas-Yedid, V., Pankajakshan, P., Lecomte, T., Montagner, Y.L., et al. (2012). Icy: an open bioimage informatics platform for extended reproducible research. Nat. Methods *9*, 690.

Desbordes, S.C., Placantonakis, D.G., Ciro, A., Socci, N.D., Lee, G., Djaballah, H., and Studer, L. (2008). High-throughput screening assay for the identification of compounds regulating self-renewal and differentiation in human embryonic stem cells. Cell Stem Cell *2*, 602–612.

Dougherty, M.T., Folk, M.J., Zadok, E., Bernstein, H.J., Bernstein, F.C., Eliceiri, K.W., Benger, W., and Best, C. (2009). Unifying biological image formats with HDF5. Commun. ACM *52*, 42–47.

Dürr, O., and Sick, B. (2016). Single-cell phenotype classification using deep convolutional neural networks. J. Biomol. Screen. *21*, 998–1003.

Eliceiri, K.W., Berthold, M.R., Goldberg, I.G., Ibáñez, L., Manjunath, B.S., Martone, M.E., Murphy, R.F., Peng, H., Plant, A.L., Roysam, B., et al. (2012). Biological imaging software tools. Nat. Methods *9*, 697–710.

Eulenberg, P., Köhler, N., Blasi, T., Filby, A., Carpenter, A.E., Rees, P., Theis, F.J., and Wolf, F.A. (2017). Reconstructing cell cycle and disease progression using deep learning. Nat. Commun. *8*, 463.

Frank, E., Hall, M.A., and Witten, I.H. (2016). The WEKA Workbench. Data Mining: Practical Machine Learning Tools and Techniques, Fourth Edition (Morgan Kaufmann), Appendix B.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Ann. Stat. *28*, 337–407.

Fusco, L., Lefort, R., Smith, K., Benmansour, F., Gonzalez, G., Barillari, C., Rinn, B., Fleuret, F., Fua, P., and Pertz, O. (2016). Computer vision profiling of neurite outgrowth dynamics reveals spatiotemporal modularity of Rho GTPase signaling. J. Cell Biol. *212*, 91–111.

Giuliano, K., DeBiasio, R., Dunlay, R.T., Gough, A., Volosky, J., Zock, J., Pavlakis, G., and Taylor, D.L. (1997). High content screening: a new approach to easing key bottlenecks in the drug discovery process. J. Biomol. Screen. *2*, 249–259.

Goldsborough, P., Pawlowski, N., Caicedo, J.C., Singh, S., and Carpenter, A. (2017). CytoGAN: generative modeling of cell images. bioRxiv. https://doi.org/10.1101/227645.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative adversarial nets. Proceeding of the Advances in Neural Information Processing Systems conference (NIPS 2014), pp. 2672–2680.

Gustafsdottir, S.M., Ljosa, V., Sokolnicki, K.L., Wilson, J.A., Walpita, D., Kemp, M.M., Seiler, K.P., Carrel, H.A., Golub, T.R., Schreiber, S.L., et al. (2013). Multiplex cytological profiling assay to measure diverse cellular states. PLoS One *8*, e80999.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR 2016), pp. 770–778.

Held, M., Schmitz, M.H., Fischer, B., Walter, T., Neumann, B., Olma, M.H., Peter, M., Ellenberg, J., and Gerlich, D.W. (2010). CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. Nat. Methods *7*, 747–754.

Horvath, P., Wild, T., Kutay, U., and Csucs, G. (2011). Machine learning improves the precision and robustness of high-content screens: using nonlinear multiparametric methods to analyze screening results. J. Biomol. Screen. *16*, 1059–1067.

Houle, D., Govindaraju, D.R., and Omholt, S. (2010). Phenomics: the next challenge. Nat. Rev. Genet. *11*, 855.

Huber, W., Carey, J.V., Gentleman, R., Anders, S., Carlson, M., Carvalho, S.B., Bravo, C.H., Davis, S., Gatto, L., Girke, T., et al. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nat. Methods *12*, 115–121.

Johnson, G.R., Donovan-Maiye, R.M., and Maleckar, M.M. (2017). Generative modeling with conditional autoencoders: building an integrated cell. arXiv, 1705.00092.

Jones, T.R., Kang, I.H., Wheeler, D.B., Lindquist, R.A., Papallo, A., Sabatini, D.M., Golland, P., and Carpenter, A.E. (2008). CellProfiler analyst: data exploration and analysis software for complex image-based screens. BMC Bioinformatics *9*, 482.

Jones, T.R., Carpenter, A.E., Lamprecht, M.R., Moffat, J., Silver, S., Grenier, J., Root, D., Golland, P., and Sabatini, D.M. (2009). Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. Proc. Natl. Acad. Sci.USA *106*, 1826–1831.

Kankaanpää, P., Paavolainen, L., Tiitta, S., Karjalainen, M., Päivärinne, J., Nieminen, J., Marjomaki, V., Heino, J., and White, D.J. (2012). BioImageXD: an open, general-purpose and high-throughput image-processing platform. Nat. Methods *9*, 683.

Kamentsky, L., Jones, T.R., Fraser, A., Bray, M.A., Logan, D.J., Madden, K.L., Ljosa, V., Rueden, C., Eliceiri, K.W., and Carpenter, A.E. (2011). Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. Bioinformatics *27*, 1179–1180.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. Proceeding of the Advances in Neural Information Processing Systems conference (NIPS 2012), pp. 1097–1105.

Kanehisa, M., and Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. *28*, 27–30.

Kraus, O.Z., Ba, J.L., and Frey, B.J. (2016). Classifying and segmenting microscopy images with deep multiple instance learning. Bioinformatics *32*, i52–i59.

Kraus, O.Z., Grys, B.T., Ba, J., Chong, Y., Frey, B.J., Boone, C., and Andrews, B.J. (2017). Automated analysis of high-content microscopy data with deep learning. Mol. Syst. Biol. *13*, 924.

Laksameethanasan, D., Tan, R.Z., Toh, G.W.L., and Loo, L.H. (2013). cellXpress: a fast and user-friendly software platform for profiling cellular phenotypes. BMC Bioinformatics *14*, S4.

Link, W., Oyarzabal, J., Serelde, B.G., Albarran, M.I., Rabal, O., Cebriá, A., Alfonso, P., Fominaya, J., Renner, O., Peregrino, S., et al. (2009). Chemical interrogation of FOXO3a nuclear translocation identifies potent and selective inhibitors of phosphoinositide 3-kinases. J. Biol. Chem. *284*, 28392–28400.

Liu, S., Zheng, W., Wu, K., Lin, Y., Jia, F., Zhang, Y., Wang, Z., Luo, Q., Zhao, Y., and Wang, F. (2017). Correlated mass spectrometry and confocal microscopy imaging verifies the dual-targeting action of an organoruthenium anticancer complex. Chem. Commun. *53*, 4136–4139.

Loo, L.H., Wu, L.F., and Altschuler, S.J. (2007). Image-based multivariate profiling of drug responses from single cells. Nat. Methods *4*, 445–453.

Ljosa, V., Sokolnicki, K.L., and Carpenter, A.E. (2012). Annotated high-throughput microscopy image sets for validation. Nat. Methods *9*, 637.

Lucchi, A., Márquez-Neila, P., Becker, C., Li, Y., Smith, K., Knott, G., and Fua, P. (2015). Learning structured models for segmentation of 2-D and 3-D imagery. IEEE Trans. Med. Imaging *34*, 1096–1110.

Marée, R., Rollus, L., Stévens, B., Hoyoux, R., Louppe, G., Vandaele, R., Begon, J.M., Kainz, P., Geurts, P., and Wehenkel, L. (2016a). Collaborative analysis of multi-gigapixel imaging data using Cytomine. Bioinformatics *32*, 1395–1401.

Marée, R., Geurts, P., and Wehenkel, L. (2016b). Towards generic image classification using tree-based learning: an extensive empirical study. Pattern Recognit. Lett. *74*, 15–23.

Masyuko, R., Lanni, E.J., Sweedler, J.V., and Bohn, P.W. (2013). Correlated imaging - a grand challenge in chemical analysis. Analyst *138*, 1924–1939.

Medical Image Net. (n.d.). http://langlotzlab.stanford.edu/projects/medical-image-net/.

Misselwitz, B., Strittmatter, G., Periaswamy, B., Schlumberger, M.C., Rout, S., Horvath, P., Kozak, K., and Hardt, W.D. (2010). Enhanced CellClassifier: a multi-class classification tool for microscopy images. BMC Bioinformatics *11*, 30.

Murphy, R.F. (2012). CellOrganizer: image-derived models of subcellular organization and protein distribution. Methods Cell Biol. *110*, 179–193.

Nunez-Iglesias, J., Kennedy, R., Parag, T., Shi, J., and Chklovskii, D.B. (2013). Machine learning of hierarchical clustering to segment 2D and 3D images. PLoS One *8*, e71715.

Nüsslein-Volhard, C., and Wieschaus, E. (1980). Mutations affecting segment number and polarity in *Drosophila*. Nature *287*, 795–801.

Ogier, A., and Dorval, T. (2012). HCS-Analyzer: open source software for high-content screening data correction and analysis. Bioinformatics *28*, 1945–1946.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2015). Is object localization for free? Weakly supervised learning with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 685–694.

Orlov, N., Shamir, L., Macura, T., Johnston, J., Eckley, D.M., and Goldberg, I.G. (2008). WND-CHARM: multi-purpose image classification using compound image transforms. Pattern Recognit. Lett. *29*, 1684–1693.

Orvedahl, A., Sumpter, R., Jr., Xiao, G., Ng, A., Zou, Z., Tang, Y., Narimatsu, M., Gilpin, C., Sun, Q., Roth, M., et al. (2011). Image-based genome-wide siRNA screen identifies selective autophagy factors. Nature *480*, 113.

Palazzolo, G., Horvath, P., and Zenobi-Wong, M. (2012). The flavonoid isoquercitrin promotes neurite elongation by reducing RhoA activity. PLoS One *7*, e49979.

Palmer, A., Phapale, P., Chernyavsky, I., Lavigne, R., Fay, D., Tarasov, A., Kovalev, V., Fuchser, J., Nikolenko, S., Pineau, C., et al. (2017). FDR-controlled metabolite annotation for high-resolution imaging mass spectrometry. Nat. Methods *14*, 57–60.

Pau, G., Fuchs, F., Sklyar, O., Boutros, M., and Huber, W. (2010). EBImage—an R package for image processing with applications to cellular phenotypes. Bioinformatics *26*, 979–981.

Pau, G., Zhang, X., Boutros, M. and Huber, W. (2018). imageHTS: Analysis of high-throughput microscopy-based screens. R package version 1.28.1.

Pärnamaa, T., and Parts, L. (2017). Accurate classification of protein subcellular localization from high-throughput microscopy images using deep learning. G3 (Bethesda) *7*, 1385–1392.

Pawlowski, N., Caicedo, J.C., Singh, S., Carpenter, A.E., and Storkey, A. (2016). Automating morphological profiling with generic deep convolutional networks. bioRxiv. https://doi.org/10.1101/085118.

Pereira, D.A., and Williams, J.A. (2007). Origin and evolution of high throughput screening. Br. J. Pharmacol. *152*, 53–61.

Piccinini, F., Balassa, T., Szkalisity, A., Molnar, C., Paavolainen, L., Kujala, K., Buzas, K., Sarazova, M., Pietiainen, V., Kutay, U., et al. (2017). Advanced cell classifier: user-friendly machine-learning-based software for discovering phenotypes in high-content imaging data. Cell Syst. *4*, 651–655.

Rämö, P., Sacher, R., Snijder, B., Begemann, B., and Pelkmans, L. (2009). CellClassifier: supervised learning of cellular phenotypes. Bioinformatics *524*, 3028–3030.

Rämö, P., Drewek, A., Arrieumerlou, C., Beerenwinkel, N., Ben-Tekaya, H., Cardel, B., and Dehio, C. (2014). Simultaneous analysis of large-scale RNAi screens for pathogen entry. BMC Genomics *15*, 1162.

Rajaram, S., Pavie, B., Wu, L.F., and Altschuler, S.J. (2012). PhenoRipper: software for rapidly profiling microscopy images. Nat. Methods *9*, 635–637.

Razi, M., and Tooze, S.A. (2009). Correlative light and electron microscopy. Methods Enzymol. *452*, 261–275.

Ribeiro, M.T., Singh, S., and Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD), pp. 1135–1144.

Rimon, N., and Schuldiner, M. (2011). Getting the whole picture: combining throughput with content in microscopy. J. Cell Sci. *124*, 3743–3751.

Robertson, S., Azizpour, H., Smith, K., and Hartman, J. (2017). Digital image analysis in breast pathology – from image processing techniques to artificial intelligence. Transl. Res. *194*, 19–35.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. Proceedings of the international conference on medical image computing and computer-assisted intervention (MICCAI 2015), pp. 234–241.

Rozenblatt-Rosen, O., Stubbington, M.J., Regev, A., and Teichmann, S.A. (2017). The human cell atlas: from vision to reality. Nat. News *550*, 451.

Sadanandan, S.K., Ranefall, P., Le Guyader, S., and Wählby, C. (2017a). Automated training of deep convolutional neural networks for cell segmentation. Sci. Rep. *7*, 7860.

Sadanandan, S., Karlsson, J., and Wählby, C. (2017b). Spheroid segmentation using multiscale deep adversarial networks. Proceedings of the IEEE international conference on computer vision workshops (ICCV 2017), pp. 36–41.

Sailem, H., Arias-Garcia, M., Bakal, C., Zisserman, A., and Rittscher, J. (2017). Discovery of rare phenotypes in cellular images using weakly supervised deep learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), pp. 49–55.

Schie, I.W., and Huser, T. (2013). Methods and applications of Raman microspectroscopy to single-cell analysis. Appl. Spectrosc. *67*, 813–828.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. Nat. Methods *9*, 676–682.

Shamir, L., Delaney, J.D., Orlov, N., Eckley, D.M., and Goldberg, I.G. (2010). Pattern recognition software and techniques for biological image analysis. PLoS Comput. Biol. *6*, e1000974.

Singh, S., Carpenter, A.E., and Genovesio, A. (2014). Increasing the content of high-content screening: an overview. J. Biomol. Screen. *19*, 640–650.

Smith, E., and Dent, G. (2005). Modern Raman Spectroscopy: A Practical Approach (John Wiley), pp. 1–210.

Smith, K., and Horvath, P. (2014). Active learning strategies for phenotypic profiling of high-content screens. J. Biomol. Screen. *19*, 685–695.

Sommer, C., and Gerlich, D.W. (2013). Machine learning in cell biology–teaching computers to recognize phenotypes. J. Cell Sci. *126*, 5529–5539.

Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.A. (2011). Ilastik: Interactive learning and segmentation toolkit. In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE, pp. 230-233.

Sommer, C., Hoefler, R., Samwer, M., and Gerlich, D.W. (2017). A deep learning and novelty detection framework for rapid phenotyping in high-content screening. bioRxiv. https://doi.org/10.1091/mbc.e17-05-0333.

Taubman, D., and Marcellin, M. (2012). JPEG2000 Image Compression Fundamentals, Standards and Practice: Image Compression Fundamentals, Standards and Practice (Springer Science and Business Media), p. 642.

Thomsen, W., Frazer, W.J., and Unett, D. (2005). Functional assays for screening GPCR targets. Curr. Opin. Biotechnol. *16*, 655–665.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., and Rinn, J.L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat. Biotechnol. *32*, 381–386.

Uhlmann, V., Singh, S., and Carpenter, A.E. (2016). CP-CHARM: segmentation-free image classification made accessible. BMC Bioinformatics *17*, 51.

Usaj, M.M., Styles, E.B., Verster, A.J., Friesen, H., Boone, C., and Andrews, B.J. (2016). High-content screening for quantitative cell biology. Trends Cell Biol. *26*, 598–611.

Venter, J.C., and Zhu, X. (2001). The sequence of the human genome. Science *291*, 1304–1351.

Wagner, M. (2009). Single-cell ecophysiology of microbes as revealed by Raman microspectroscopy or secondary ion mass spectrometry imaging. Annu. Rev. Microbiol. *63*, 411–429.

Weigert, M., Schmidt, U., Boothe, T., Andreas, M., Dibrov, A., Jain, A., and Myers, E.W. (2017). Content-aware image restoration: pushing the limits of fluorescence microscopy. bioRxiv. https://doi.org/10.1101/236463.

Wieslander, H. and Forslid, G. (2017). Deep convolutional neural networks for detecting cellular changes due to malignancy. Proceedings of the IEEE International Conference on Computer Vision workshops (ICCV 2017), pp. 82–89.

Winsnes, C.F., Sullivan, D.P., Smith, K., and Lundberg, E. (2016). Multi-label prediction of subcellular localization in confocal images using deep neural networks. Mol. Biol. Cell *27*, 26–39.

Yamauchi, Y., Boukari, H., Banerjee, I., Sbalzarini, I.F., Horvath, P., and Helenius, A. (2011). Histone deacetylase 8 is required for centrosome cohesion and influenza A virus entry. PLoS Pathog. *7*, e1002316.

Young, D.W., Bender, A., Hoyt, J., McWhinnie, E., Chirn, G.W., Tao, C.Y., Tallarico, J.A., Labow, M., Jenkins, J.L., Mitchison, T.J., and Feng, Y. (2008). Integrating high-content screening and ligand-target prediction to identify mechanism of action. Nat. Chem. Biol. *4*, 59.

Zanella, F., Lorens, J.B., and Link, W. (2010). High content screening: seeing is believing. Trends Biotechnol. *28*, 237–245.

Zhang, X., and Boutros, M. (2013). A novel phenotypic dissimilarity method for image-based high-throughput screens. BMC Bioinformatics *14*, 336.

Zhang, H., Xie, X., Fang, C., Yang, Y., Jin, D., and Fei, P. (2018). High-throughput, high-resolution generated adversarial network microscopy. arXiv, 1801.07330.