Machine learning in cell biology – teaching computers to recognize phenotypes

Christoph Sommer and Daniel W. Gerlich*

Institute of Molecular Biotechnology of the Austrian Academy of Sciences (IMBA), 1030 Vienna, Austria

*Author for correspondence (daniel.gerlich@imba.oeaw.ac.at)

Journal of Cell Science 126, 1–11
© 2013. Published by The Company of Biologists Ltd
doi: 10.1242/jcs.123604

Summary

Recent advances in microscope automation provide new opportunities for high-throughput cell biology, such as image-based screening. High-complex image analysis tasks often make the implementation of static and predefined processing rules a cumbersome effort. Machine-learning methods, instead, seek to use intrinsic data structure, as well as the expert annotations of biologists to infer models that can be used to solve versatile data analysis tasks. Here, we explain how machine-learning methods work and what needs to be considered for their successful application in cell biology. We outline how microscopy images can be converted into a data representation suitable for machine learning, and then introduce various state-of-the-art machine-learning algorithms, highlighting recent applications in image-based screening. Our Commentary aims to provide the biologist with a guide to the application of machine learning to microscopy assays and we therefore include extensive discussion on how to optimize experimental workflow as well as the data analysis pipeline.

Key words: Bioimage informatics, Computer vision, High-content screening, Machine learning, Microscopy

Introduction

Commercially available motorized microscopes can yield data at a throughput of $>10^5$ images per day, raising a strong need for automated data analysis (Conrad and Gerlich, 2010; Lock and Strömblad, 2010). Computational data analysis not only reduces the workload for the experimentalist, but also ensures objectivity and consistency in the annotation of large data sets (Danuser, 2011). The complexity and diversity in microscopic image data, however, poses challenges for developing suitable data analysis workflows.

Bioimage informatics methods offer powerful solutions for specific image analysis tasks, such as object detection, motion analysis or measurements of morphometric features (Danuser, 2011; Murphy, 2011; Eliceiri et al., 2012; Myers, 2012). Most image analysis algorithms, however, have been developed for specific biological assays. The application of the respective algorithms to other markers or cell types then often requires parameter tuning or even re-programming of the software. Manual software adaptations, however, are tedious and provide major obstacles for most cell biological laboratories, owing to the limited knowledge about the mathematics behind the image analysis algorithms and a lack of expertise in software engineering.

Machine learning aims to provide a general solution to this problem by learning processing rules from examples rather than relying on manual adjustments of parameters or pre-defined processing steps (Hastie et al., 2005; Bishop, 2006; Domingos, 2012). Machine learning is particularly superior to conventional image processing programs when it comes to solving complex multi-dimensional data analysis tasks such as discriminating morphologies that are not easily described by a few parameters (Boland and Murphy, 2001; Conrad et al., 2004; Neumann et al., 2010).

Machine learning generally proceeds in two phases (Hastie et al., 2005; Bishop, 2006). In the training phase, a collection of data samples is used to build or improve a computer system by learning from inherent structure and relationships within this data. This computer system is then applied to new data samples to predict certain properties of these data samples. Thus, the overall goal of any machine-learning method is to generalize from a few training examples to make accurate predictions on large sets of data samples that were not observed during training (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; de Ridder et al., 2013).

A common machine-learning discipline is classification. In this approach, the user generates a training data set by annotating some representative examples according to predefined classes. The machine-learning algorithm automatically infers the rules to discriminate the classes, which can then be applied to the full data set. This type of learning is termed ‘supervised’ machine learning, and its principal goal is to infer general properties of the data distribution from a few annotated examples (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; de Ridder et al., 2013). Supervised machine learning has been successfully applied in diverse biological disciplines, such as high-content screening (Kittler et al., 2004; Lansing Taylor et al., 2007; Doil et al., 2009; Collinet et al., 2010; Fuchs et al., 2010; Neumann et al., 2010; Schmitz et al., 2010; Mercer et al., 2012), drug development (Perlman et al., 2004; Slack et al., 2008; Loo et al., 2009; Castoreno et al., 2010; Murphy, 2011), DNA sequence analysis (Castelo and Guigó, 2004; Ben-Hur et al., 2008) and proteomics (Yang and Chou, 2004; Datta and Pihur, 2010; Reiter et al.,

JCS Advance Online Article. Posted on 20 November 2013

2011), as well as in many other fields outside of biology, such as speech (Rabiner, 1989) and face recognition (Viola and Jones, 2004), and prediction of stock market trends (Kim, 2003).

A second type of machine learning extracts information from the data completely independently of user annotations. The goal of 'unsupervised' machine learning is to group data points into clusters on the basis of a similarity measure or to facilitate data mining by reducing the complexity of the data (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; de Ridder et al., 2013). Unlike supervised approaches, unsupervised methods enable the exploration of unknown phenotypes (Wang et al., 2008; Lin et al., 2010) and have been successfully used for phenotypic profiling of drug effects (Perlman et al., 2004).

A number of recent reviews and textbooks provide extensive theoretical background on different machine-learning algorithms (Hastie et al., 2005; Bishop, 2006; Larrañaga et al., 2006; Tarca et al., 2007; Danuser, 2011; de Ridder et al., 2013). Successful application of machine learning, however, also needs to take into account many practical considerations and it requires knowledge about the specific data type and analysis goals. This Commentary aims to provide a guide for the cell biologist to establish an efficient machine-learning pipeline for the analysis of microscopic images. We first discuss how image data are converted into units that serve as input for machine-learning methods. We then provide background on state-of-the-art supervised machine-learning methods and discuss what needs to be taken into account to optimize their performance. We also introduce the basic concepts of unsupervised machine learning and highlight some recent applications in cell biology.

The machine-learning pipeline for cell phenotyping

Machine learning is widely used in image-based screening to classify cell morphologies that are traced by fluorescent markers. The principal objective of the screening is to determine whether an experimental perturbation (e.g. treatment with a chemical compound, small interfering RNA or genetic manipulation) leads to a cellular phenotype (e.g. change in cell morphology, protein expression level or anything that can be probed by imaging biosensors). The most commonly used machine-learning method, classification, is based on the definition of phenotypes by representative examples (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; de Ridder et al., 2013). Thus, before a screen can be conducted, examples need to be recorded for unperturbed negative controls as well as for expected classes of phenotypes. If representative examples for phenotypes are not available and cannot be obtained, supervised machine learning is not applicable and unsupervised methods need to be used instead (see below).

The actual machine-learning algorithm is typically embedded into a processing pipeline that converts original raw data into units that are suitable as input for the respective machine-learning algorithm (Tarca et al., 2007; de Ridder et al., 2013). The principal input for any learning algorithm is a set of objects, each of which are described by quantitative features. For cell biological applications based on microscopy data, the typical processing pipeline comprises image pre-processing, object detection and feature extraction (Fig. 1).

Image pre-processing

The first step of the machine-learning pipeline, image pre-processing, aims to remove artifacts produced by the microscope or camera. For instance, uneven illumination of the microscope field of view should be compensated for by image flat-field correction (Buchser et al., 2004). This normalizes the cellular signal intensity levels, as these should not change with the position inside the imaging field. Pixel noise resulting from low light exposure, particularly in live-cell imaging applications, should also be removed by smoothing filters (Lindblad et al., 2004). In time-lapse movies, subsequent images might not be in the registry owing to a random or systematic drift of the microscope stage position. Image registration techniques find optimal image transformations to correct for such artifacts (Thévenaz et al., 1998; Oliveira and Tavares, 2012).

Object detection

Next, the objects of interest, which form the basis for classification, need to be defined. Most machine-learning pipelines separate objects of interest (e.g. cells) from image background, yet classification can also be performed at the level of image pixels (Kaynig et al., 2010; Sommer et al., 2011) or whole unsegmented images (Huang and Murphy, 2004; Shamir et al., 2008; Weber et al., 2013) (Fig. 2). Object detection is either based on region properties (e.g. bright regions can be segmented from background by intensity thresholding), or based

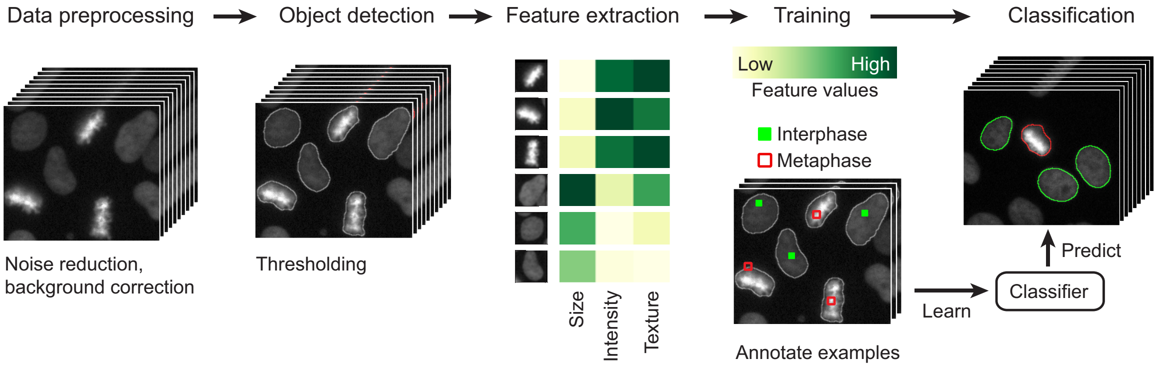

Fig. 1. The machine-learning pipeline for analysis of microscopy data. The canonical processing steps include image preprocessing, object detection, feature extraction, classifier training and classification. Classifier training yields a classifier by learning from a representative set of annotated training examples. Afterwards, new objects are automatically predicted in the classification phase. The image data used for illustration of the workflow shows live human HeLa cells stably expressing a fluorescently labeled chromatin marker [core histone 2B fused to mCherry (Held et al., 2010)].

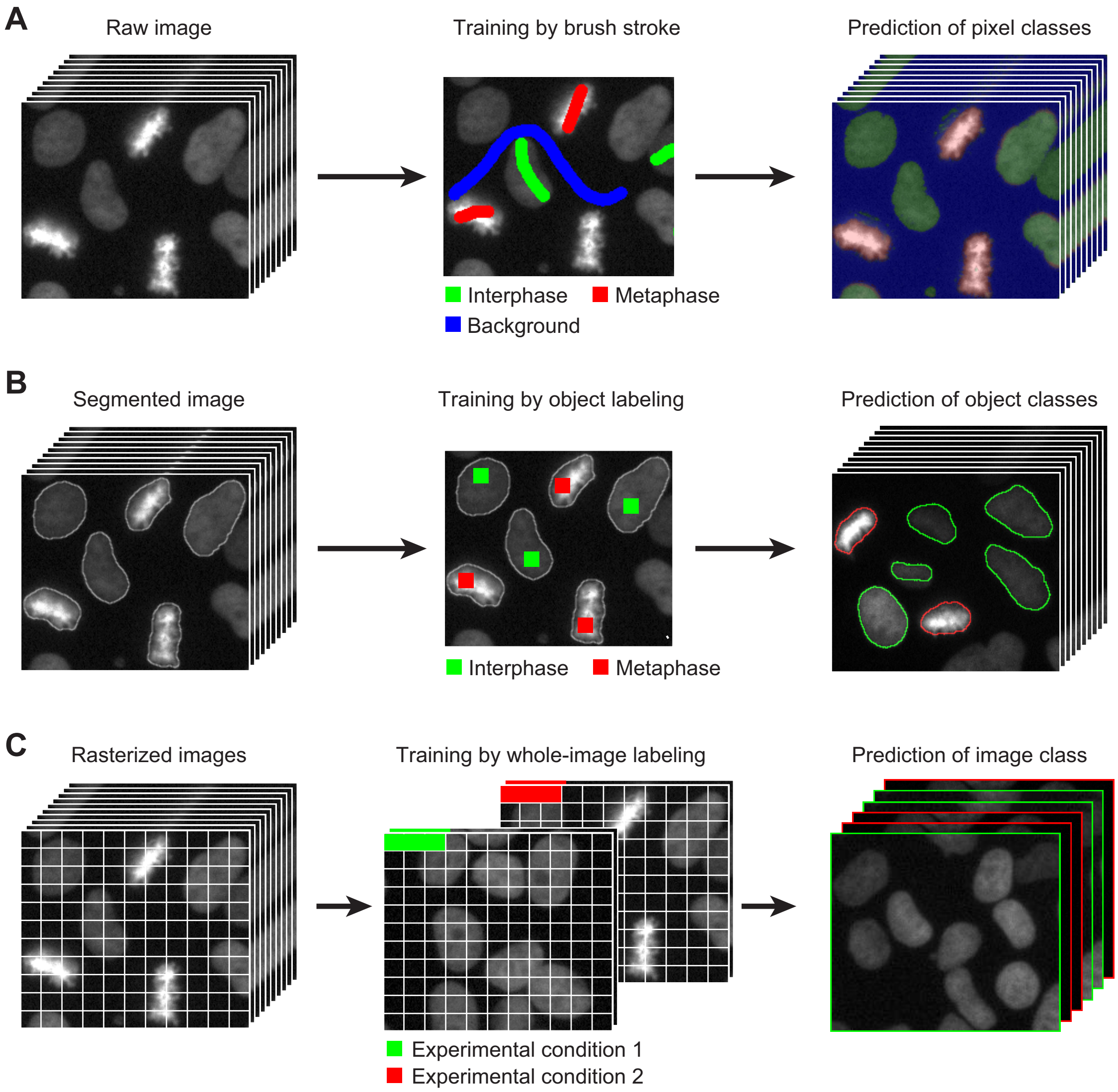
A
Raw image
Training by brush stroke
Prediction of pixel classes
🟩 Interphase
🟥 Metaphase
🟦 Background
B
Segmented image
Training by object labeling
Prediction of object classes
🟩 Interphase
🟥 Metaphase
C
Rasterized images
Training by whole-image labeling
Prediction of image class
🟩 Experimental condition 1
🟥 Experimental condition 2

Fig. 2. Image classification by supervised machine learning at the level of pixels, cell objects or whole images. Image data shows human HeLa cells expressing a chromatin marker as in Fig. 1. (A) Pixel classification for image segmentation using ilastik (Sommer et al., 2011). Pixels of cells and background regions are annotated interactively by brush strokes according to pre-defined classes. Features of the labeled pixels and their local neighborhood are then used to learn a pixel classifier. Afterwards, this classifier is used to predict new images in a pixel-wise fashion to obtain a partitioning of the image into the phenotype classes. (B) Object classification for analysis of cellular phenotypes with CellCognition (Held et al., 2010). Each segmented cell is user-labeled according to its cell cycle state in order to learn a classifier, which is then applied to unseen data to predict cell morphology classes. Total accuracies of >95% can be achieved by this approach, such as in the discrimination of eight different cell cycle stages based on a chromatin marker (Held et al., 2010). Similar approaches have been used to screen for DNA damage response signaling (Doil et al., 2009) and to classify subcellular protein localization (Boland and Murphy, 2001). (C) Segmentation-free image classification by Wndchrm software (Shamir et al., 2008). Image features characterize the image as a whole and classification outputs a class membership per image. Segmentation-free approaches are applied in cases in which segmentation of objects is difficult or impossible owing to high cell densities (cells are touching) or when dealing with complex cellular structures, such as dendrites of neuronal cells (Weber et al., 2013).

on contours (e.g. edges can be detected based on the local image gradient). No single method, however, is suitable to solve all possible segmentation problems in cell-based screening, and it is therefore inherently difficult to generalize the image segmentation method. The segmentation of the image can also be facilitated by machine learning: pixel classifiers that work on local pixel neighborhoods aim to learn to separate foreground (e.g. cells) from background by classifying whether pixels belongs to an object (Tu and Bai, 2010; Sommer et al., 2011).

To ease the image segmentation task, many imaged-based screening projects use reference markers such as fluorescent chromatin or DNA labels (Kittler et al., 2007; Collinet et al., 2010; Neumann et al., 2010; Schmitz et al., 2010; Mercer et al., 2012). On the basis of the primary segmentation marker, secondary object regions can be derived in order to probe diverse secondary markers without the need to adapt the program code for segmentation of the secondary image channel.

When analysis on a single-cell level is not required, it is possible to apply machine learning on unsegmented images

(Fig. 2C) by classifying image features that do not require object segmentation (Huang and Murphy, 2004) or by learning phenotypic distances based on rectangular image patches (Hamilton et al., 2009; Rajaram et al., 2012a).

Feature extraction

Following segmentation, each object needs to be described by quantitative features that form the basis to distinguish them by a classifier algorithm. The performance of a machine-learning pipeline relies substantially on an appropriate collection of relevant features (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; de Ridder et al., 2013). The raw image pixel intensities are not well suited as features, because they withhold information on spatial and spectral patterns and can contain undesirable information such as the absolute orientation of cell objects (Huh et al., 2009). Thus, descriptive features need to be derived from the pixel intensities that enrich information relevant for classification.

Two types of features are widely used to describe cell objects in microscopic images. Texture features quantify the distribution of pixel intensities within each object. Simple examples are mean intensity and standard deviation. More advanced texture features measure the granularity at different scales (Chen et al., 1995; Chebira et al., 2007) or pixel–pixel co-occurrence patterns (Haralick, 1979). A second class of feature describes the contour on the basis of the segmentation mask, for example, the contour roughness or circularity (Liu et al., 2011). Many powerful morphometric features are abstract representations of images and therefore difficult to intuitively relate to visual inspection of the cell image. Relevant features that relate to a phenotype can be automatically determined by the learning algorithm, and will vary with the specific biological marker and assay (Fig. 3). To avoid tedious manual adaptations of feature sets for each specific application, multi-purpose feature libraries have been developed, and these cover the needs for most cell biological assays (Jones et al., 2008; Held et al., 2010; Shariff et al., 2010).

Even though a versatile applicability of a machine-learning pipeline requires comprehensive feature sets, gathering more features does not always improve performance. This is because the increase in dimensionality with each feature renders the classification task exponentially more complex. This is referred to as the 'curse of dimensionality' (Hastie et al., 2005) and can be addressed by algorithms that reduce dimensionality, for example, by selecting the most informative features (Loo et al., 2007; Saeys et al., 2007). Engineering the right set of features is often key to the success of a machine-learning project, and at least as important as using the right learning algorithm (Fig. 4).

In summary, the processing pipeline yields a set of objects (typically representing cells), each of which is associated with an ordered list of feature values called the feature vector. Objects are thus represented in a multi-dimensional feature space, where the number of features defines the dimensionality. The challenging task of supervised machine learning is then to infer rules for how to

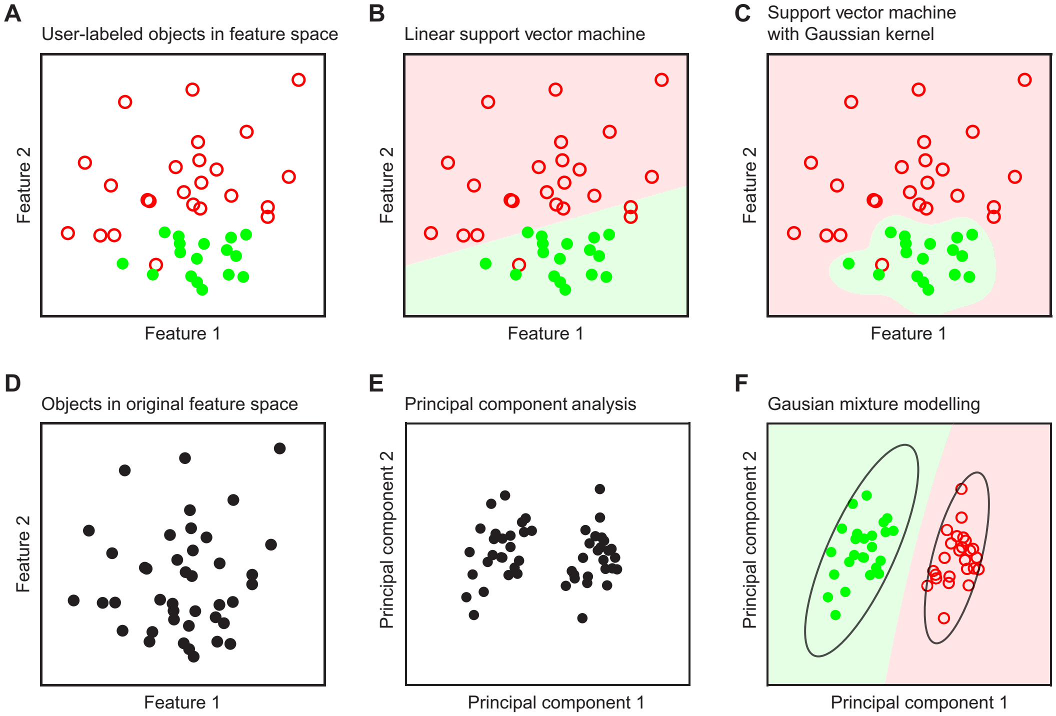

Fig. 3. Object representation in feature space. Data points are extracted from image data as shown in Figs 1 and 2 (Held et al., 2010**).** (A–C) Supervised machine learning. (A) Each dot represents one cell object of representative metaphase (red) or interphase cells (green), as shown in Fig. 1. The two feature dimensions have been arbitrarily chosen from a full set of 239 features (Held et al., 2010). (B) A decision boundary between interphase (green area) and metaphase (red area) cells was derived by a linear support vector machine based on the labeled training objects. (C) As in B, but using a non-linear support vector machine with a Gaussian kernel. (D–F) Unsupervised machine learning. (D) Each dot represents one cell object of the same data shown in A, but metaphase or interphase was not annotated by the user. (E) The original set of 239 features was transformed by principal component analysis. The same objects shown in A–D are plotted on the axes of principal components 1 and 2. (F) The cell objects shown in D and E were clustered by Gaussian mixture models (Bishop, 2006) on the first two principal components. Red and green indicate the two different cluster labels.

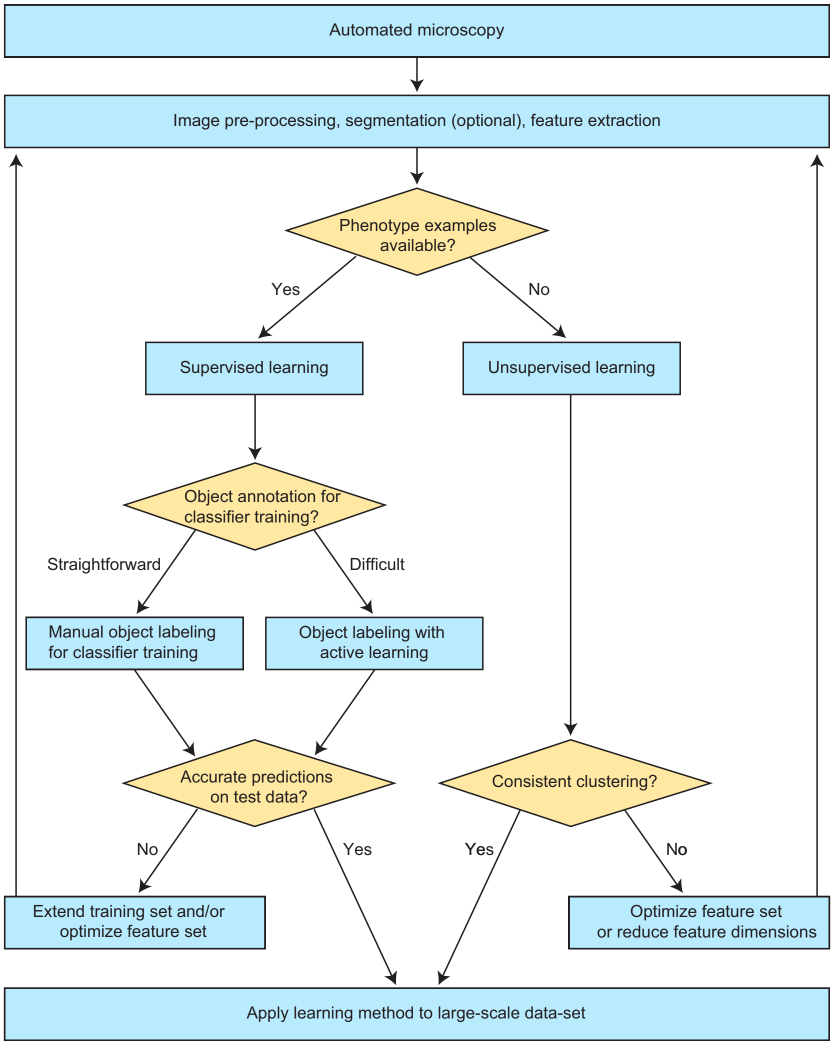
Automated microscopy
Image pre-processing, segmentation (optional), feature extraction
Phenotype examples available?
Yes
No
Supervised learning
Unsupervised learning
Object annotation for classifier training?
Straightforward
Difficult
Manual object labeling for classifier training
Object labeling with active learning
Accurate predictions on test data?
Consistent clustering?
No
Yes
Yes
No
Extend training set and/or optimize feature set
Optimize feature set or reduce feature dimensions
Apply learning method to large-scale data-set

Fig. 4. Implementing and optimizing a machine-learning pipeline in image-based screening.

discriminate different classes of objects in this multi-dimensional feature space.

How does a machine learn?

As discussed above, there are two different types of machine learning, supervised and unsupervised learning. Supervised learning is guided by user training with the goal of subsequently applying a learned program to a similar task on independent large data sets (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; Domingos, 2012; de Ridder et al., 2013). Unsupervised learning, by contrast, is fully independent of user interaction and aims to recognize patterns in the data to facilitate the interpretation of complex multi-dimensional data (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; de Ridder et al., 2013). As supervised machine learning has been used much more widely in cell biology, we focus our Commentary on this approach and only outline general aspects of unsupervised methods at the end of this section.

Supervised machine learning: learning from user-defined examples

In supervised machine learning, a human expert first defines the processing task by annotating a small subset of objects from the original data set, for example, by phenotype labels according to cell morphology (Fig. 1). This training data serves to automatically infer internal parameters of a learning model (the learner), which is then applied to discriminate between the different classes of objects in the full data set. Thus, the overall task of supervised machine learning is to generalize from a few selected examples.

The supervised learning process is guided by an objective function, which evaluates how well the learner adapted to the training data (Hastie et al., 2005; Bishop, 2006; Domingos, 2012). On the basis of the objective function, an optimization procedure seeks parameters that yield the best learner. Importantly, the overall goal is to obtain a learner that generalizes: the learner needs to perform well on data that was not used for training. It is therefore essential to withhold a fraction of the training data to test this. If the learning performance were only evaluated based on the data used for learning, a simple memorization of the examples might perform best, which is likely to yield poor results on independent data. Various strategies have been developed for optimal splitting of training data into fractions that serve learning and testing, respectively (see below).

Supervised machine learning has been an important backbone for analysis pipelines in many high-content screening projects

(Kittler et al., 2007; Fuchs et al., 2010; Neumann et al., 2010; Schmitz et al., 2010; Mercer et al., 2012). The strengths of supervised machine learning are intuitive assay development based on examples, the versatility and applicability to diverse assays, and efficient and robust computation of large datasets. This approach, however, depends on phenotype examples, which precludes searching for novel and unexpected phenotypes in screens.

The models underlying machine-learning algorithms

How is the learning process implemented in a computer algorithm? There are two principally different types of learning models: generative approaches, which model the distribution of data points, or discriminant approaches, which model decision boundaries between different classes (Hastie et al., 2005; Bishop, 2006; for details on specific algorithms, see Box 1).

Box 1. Supervised classification algorithms

State-of-the-art supervised classification methods have been optimized towards classification accuracy, computational performance, learning from as few training objects as possible and versatility in their application. Widely used algorithms are described below.

Support vector machines
Support vector machines (SVMs) aim to find a decision hyperplane that separates data points of different classes with a maximal margin (i.e. maximal distance to the nearest training data points). Because data points of different classes might not always be completely separable by a hyperplane, most SVM implementations are based on a soft margin, which allows misclassifications at a certain cost value. SVMs themselves are linear classifiers, but they can generate non-linear decision boundaries if the data points are transformed beforehand to higher dimensions (such as a Gaussian kernel) using a mapping function (Vapnik, 2000). SVMs are relatively robust towards noisy features and are computationally efficient, and implementations are available in diverse bioimaging software packages (Held et al., 2010; Conrad et al., 2011; Horvath et al., 2011).

Adaptive boosting
Adaptive boosting (AdaBoost) combines several ‘weak’ learners to form a ‘strong’ classifier by iteratively adding and reweighting simple classifiers such as thresholds (Freund and Schapire, 1995). Owing to its iterative nature, boosting is particularly suitable for interactive online learning (Jones et al., 2008). However, AdaBoost is relatively sensitive towards noisy data and outliers (Kanamori et al., 2007). A widely used implementation, GentleBoost (Friedman et al., 2000), is available in the bioimaging software package CellProfiler Analyst (Jones et al., 2008).

Random forest
Random forests (RFs) (Breiman, 2001) train an ensemble of decision trees (Breiman et al., 1983) under random influence to average their outcome. Averaging the prediction of an ensemble reduces the overall variance while maintaining the low bias typical for decision trees. RFs are robust in high dimensions, because of an implicit feature selection, and are computationally efficient and easily parallelizable. An RF implementation widely used in cell biological applications is available (Kaynig et al., 2010; Sommer et al., 2011).

Generative methods model statistical distributions underlying the data objects. This can be based on certain probability distributions (e.g. Gaussian distributions), whose parameters are estimated from the training data (parametric models). Decision boundaries that separate data points according to their class membership are formed implicitly. Generative models can be used to synthesize new data points, which might be useful in some specialized applications [e.g. simulation of cell morphology (Buck et al., 2012; Rajaram et al., 2012b)]. Generative models have also been successfully applied to correct misclassifications of cell cycle stages, aided by temporal information in time-lapse movies (Held et al., 2010) or the discovery of new biologically active peptide hormones by searching for sequence features in protein sequences (Mirabeau et al., 2007) using hidden Markov models (Rabiner, 1989).

Discriminant approaches, by contrast, directly model the decision boundary between different classes rather than the distribution of data points. The simplest implementation is a linear decision boundary (or a hyperplane in high-dimensional feature space). Linear discriminant methods are very robust towards noise in the data, yet their decision boundaries cannot accurately discriminate objects of different classes if they are distributed in complex patterns, such as typically observed for cell morphologies (Meyer et al., 2003; Loo et al., 2007; Fuchs et al., 2010; Held et al., 2010; Neumann et al., 2010). Most discriminant methods used in cell biological applications, therefore, use non-linear classifiers, which can express more complex decision boundaries.

The complexity of non-linear decision boundaries can range from smoothly bent functions to arbitrary rugged and unconnected boundaries (Fig. 3A–C). The more complex a decision boundary, the better it can separate complex distributions of data points. By contrast, complex decision boundaries are more likely to represent details that are specific to the sampled training data or noise and therefore might not apply to the general distribution of other data points.

These characteristics of classifiers are referred to as bias and variance (Hastie et al., 2005; Bishop, 2006; Domingos, 2012). A high bias means a strong preference of the learner to follow its internal model assumptions, even if this does not match well to the training data. A linear classifier will therefore always yield a linear classification boundary even if this leads to severe misclassifications on non-linear data distributions. A low bias, by contrast, indicates that a classifier has no strong internal model assumptions and is able to adapt to arbitrary cluttered training data. A learner with the lowest bias, however, is not necessarily the optimal solution, because the ability to generalize from training data are also assessed by a second parameter termed variance.

The variance of a classifier indicates its stability when repeatedly applied to subsets of training data points drawn independently from the same underlying data source (e.g. the same biological experiment). Classifiers with a low variance produce similar decision boundaries when applied to different training sets, whereas high variance classifiers are prone to adapt to noise and particularities of that very instance of training data. A major design goal for machine-learning algorithms is to optimize the trade-off between bias and variance. In many implementations, this can be controlled by parameters whose optimal values depend on the specific experimental data.

In light of the diversity of supervised machine-learning methods, how can we identify the best algorithm? Important requirements are maximal accuracy and versatile application to diverse cell biology assays without the need to adapt software. Whether generative or discriminative classification approaches are better suited to solve a machine-learning task depends on how well internal model assumptions are met in the data (Ng and Jordan, 2002). For instance, support vector machines (discriminative approach) are widely used in cell biology (Meyer et al., 2003; Loo et al., 2007; Fuchs et al., 2010; Held et al., 2010; Neumann et al., 2010) owing to their good average performance among benchmark data sets (Meyer et al., 2003) and applicability to different data structures (Hastie et al., 2005). However, generative approaches, such as linear discriminant analysis, might be favorable in other cases, such as classifying the phenotypes of the actin cytoskeleton in *Drosophila melanogaster* cells (Wang et al., 2008).

Other considerations can be taken into account depending on the specific application. For example, methods are preferred if they require only small numbers of training objects for good performance. Some applications might require a human to interpret the decision rules of the classifier. Other applications might need a particularly fast computing performance. Some methods that have been found to be particularly versatile and powerful for cell biological applications are specified in Box 1 and software implementations are listed in Box 2.

Box 2. Machine-learning software for cell biologists

Machine learning methods have been implemented in a number of open-source software projects dedicated to high-content screening data (Shamir et al., 2010; Eliceiri et al., 2012).

CellProfiler and **CellProfiler Analyst** (Carpenter et al., 2006; Jones et al., 2008; Kamentsky et al., 2011) (http://www.cellprofiler.org). A particular strength of these software packages is a modular workflow design, which enables rapid development of analysis assays. CellProfiler Analyst provides a multi-class active learning interface based on boosting. CellProfiler runs on all major operating systems and supports computing on clusters for large-scale screening.

CellCognition (Held et al., 2010) (http://www.cellcognition.org/) has been optimized for time-resolved imaging applications. It comprises a complete machine-learning pipeline from cell segmentation and feature extraction to supervised and unsupervised learning. CellCognition runs on all major operating systems and supports computing on clusters for large-scale screening.

ilastik (Sommer et al., 2011) (http://www.ilastik.org/) is an interactive segmentation tool based on pixel classification, which facilitates more complex image-segmentation tasks and provides real-time feedback.

Bioconductor imageHTS and **EBImage** (Gentleman et al., 2004; Pau et al., 2010; Pau et al., 2013) (http://www.bioconductor.org/; http://bioconductor.org/packages/devel/bioc/html/imageHTS.html) provide a versatile toolbox for statistical data analysis and image processing in the programming language R.

PhenoRipper (Rajaram et al., 2012a) (http://www4.utsouthwestern.edu/altschulerwulab/phenoripper/) learns the phenotypic distance of cell populations without the need to segment individual cells.

Wndchrm (Shamir et al., 2008) (https://code.google.com/p/wnd-charm/) provides a command-line program for the segmentation-free classification of entire images.

Fiji (Schindelin et al., 2012) (http://fiji.sc/Fiji) is an ImageJ (Java) distribution that contains many image-analysis and machine-learning plug-ins.

CellExplorer (Long et al., 2009) provides 3D image analysis and machine-learning methods in MATLAB®.

Data format standards for high-content screening such as **CellH5** (Sommer et al., 2013) and **SDCubes** (Millard et al., 2011) aim at facilitating inter-operability between different software packages by storing multi-dimensional original image data together with processing parameters and intermediate processing results. CellH5 has interfaces to R Bioconductor (Gentleman et al., 2004) and CellCognition (Held et al., 2010), and can be natively accessed from all major programming languages; SDCubes has been implemented for **ImageRails** (Millard et al., 2011).

How to measure and optimize the performance of machine learning?

The most widely used performance metric for a learner is total error, that is, the ratio of incorrect classifications divided by the total number of objects. Depending on the learning task, it can be useful to decompose the total error into false-positive and false-negative errors, which enables specific optimization strategies. For instance, if an RNA interference screen yields a long candidate gene list that cannot be completely validated by secondary assays, it could be useful to minimize false-positive prediction of phenotypes, taking into account that some potential phenotypes might be missed. If the most important goal of a screen is comprehensiveness and it is feasible to validate all candidates by secondary analysis, then it might be preferred to minimize false-negative classifications (e.g. misclassification of a phenotype as a negative control morphology) by taking into account an increased false-positive error rate.

Accurate evaluation of the performance of a machine-learning method needs a comprehensive and representative data set for the specific goal. In light of the diversity of data types and analysis tasks in cell biology, it is often difficult to estimate the performance of published learning methods based on the specific proof-of-concept data used in the respective study. For objective benchmarking of learning methods in high-content screening, several annotated reference data sets have been published (Ljosa et al., 2012; Rajaram et al., 2012b).

How many data objects are required to train a good learner? Unfortunately, there is no general rule, because this depends on the method and the variability within the specific data set. In practice, some applications can yield satisfying results by training with ten objects per class, although most applications will require substantially more. Discriminative methods typically need more training objects to achieve a satisfactory performance than do generative models (Ng and Jordan, 2002). Irrespective of the learning algorithm, an increase in the number of features generally requires more training examples (Hastie et al., 2005). The most important evaluation criterion for a learner is its ability to generalize (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; Domingos, 2012; de Ridder et al., 2013). To measure this, the available annotated reference data needs to be split into three subsets. The first fraction of objects is used for the initial learning. A second fraction of objects serves to improve the parameter settings of the learner. Finally, the performance of the learner is evaluated against the third fraction, the independent test data. This procedure prevents overfitting and allows for a good generalization (Hastie et al., 2005; Bishop, 2006; Tarca et al., 2007; Domingos, 2012; de Ridder et al., 2013).

To make most efficient use of a limited number of training objects, a procedure termed *k*-fold cross-validation has been developed (Kohavi, 1995; Ambroise and McLachlan, 2002). The training data set is partitioned into a user-defined number of *k* subsets, of which all but one are used for initial training of the learner. The remaining fraction serves to measure the performance of the learner and optimize its parameters. This is repeated for all fractions of data, typically five or ten times.

When a specific class is highly overrepresented in the data, an optimization towards total accuracy might yield a learner that performs poorly on predicting the less-abundant classes. This problem can be tackled either by sub-sampling only a fraction of training objects from the abundant classes while preserving all training objects from the less-abundant classes, or by specialized learning algorithms (Kotsiantis et al., 2006).

What overall accuracy can we expect from machine learning in a typical cell biological experiment? This is difficult to express in absolute numbers because it depends on many different parameters and the quality of the data. Many cell biological applications have achieved total accuracies of >90%, often within the range of object labeling inconsistencies between different human annotators.

Unsupervised machine learning – learning from intrinsic data structure

In some biological applications it is difficult or impossible to define a training data set, which precludes the use of supervised machine-learning methods. For example, an image-based screen might be aimed at the discovery of a hypothetical morphological deviation that has not been observed before. In such cases, unsupervised machine-learning methods can be used to detect individual outlier objects or clusters of objects that differ from the control group in a dataset (Fig. 3D–F). The overall goal of unsupervised machine learning is the identification of structures in the input data without prior user definition of the output.

In the absence of annotated training data, the definition of an objective function becomes more difficult, as it cannot make use of classification error rates. Instead, objective functions in unsupervised learning are typically based on distances in the feature space. For instance, clustering methods aim to group objects into clusters by minimizing the distance between objects within each cluster and maximizing the distance between different cluster centers (Bishop, 2006; Box 3).

Another widely used unsupervised method is dimensionality reduction (Van der Maaten et al., 2009), which aims to find a less redundant and lower-dimensional representation of the data points, keeping as much information as possible from the original high-dimensional feature space (Fig. 3D,E). Dimensionality reduction enables better visualization of the data points and thereby facilitates data mining by visual inspection.

Despite the advantage of fully automated data analysis without user training, unsupervised learning has not yet been widely used in cell biological applications. The biggest problem is the relatively poor performance on noisy data and the unpredictable output, which limits the interpretation, particularly when the cluster differences relate to complex combinations of multiple features. To overcome these limitations, some applications of unsupervised learning have incorporated additional knowledge about the data, such as, for example, temporal constraints on morphological transitions (Zhong et al., 2012) or non-negativity constraints on gene expression data (Devarajan, 2008).

Box 3. Unsupervised machine-learning algorithms

The main disciplines of unsupervised learning are clustering and dimensionality reduction. Clustering aims at assigning categorical class labels to data points without prior training. Widely used clustering methods are described below.

k-means clustering

k-means clustering finds a user-defined number (*k*) of clusters by an iterative procedure. The cluster centers are initialized randomly and each data point is first assigned to the closest cluster center. Then, each cluster center is recalculated based on the mean of all assigned data points. This is repeated until convergence (i.e. the cluster centers) does not change beyond a significance threshold in the update step.

Gaussian mixture model

Gaussian mixture model (GMM) extends *k*-means clustering by accounting for more complex data distributions. In addition to estimating cluster centers (means), each cluster center is associated with parameters that describe a Gaussian distribution. The estimation of a variance per cluster enables the modeling of data clusters with elliptical data spread.

Hierarchical clustering

In contrast to *k*-means and GMM clustering, hierarchical clustering is directly based on distances between the data points. In the first step, all data points are defined as single clusters. Then clusters are merged according to a linkage criterion based on small distances. This process is recursively applied, yielding a hierarchical cluster tree termed a dendrogram. Hierarchical clustering has been widely used to visualize similarities between complex phenotypes and is implemented in, for example, Bioconductor (Gentleman et al., 2004).

Dimensionality reduction

Dimensionality reduction is used to facilitate visual inspection of high-dimensional data. This is necessary because data points are very scarcely distributed in the high-dimensional feature space, which grows exponentially with the number of dimensions (Hastie et al., 2005; Bishop, 2006; Domingos, 2012). Dimensionality reduction also enables a more compact and less redundant visualization of the data owing to the smaller number of features. Widely used methods for dimensionality reduction are:

- Principal component analysis (PCA), which maps original data points by a linear transformation (rotation) to a new feature space, where all transformed features are mutually uncorrelated. The resulting dimensions (principal components, PCs) are ranked by the amount of variance they cover in the data. The highest-ranked PCs thus enrich relevant information, and low-ranked PCs can be removed for further data analysis (Fig. 3). Owing to its wide applicability and effectiveness, PCA is often used for visualization and as a preprocessing step in classification and clustering.
- Independent component analysis (ICA) is related to PCA, but searches for a non-linear transformation that yields maximally independent dimensions.
- Multidimensional scaling (MDS) aims to construct a lower-dimensional mapping such that the original distances are preserved as much as possible.
- Feature selection, by contrast, does not transform the original data, but instead selects a subset of most relevant dimensions from the original space, which is easier to interpret.

Active learning – computer assists the user in data annotation

A major bottleneck in supervised learning is the generation of user-annotated labels. Human experts might introduce bias and subjective variability into the training data set if information about the true object state is unattainable (Zhong et al., 2012). In addition, it is difficult, and in many cases impossible, to anticipate the gain in learning achieved by selecting and annotating a particular data point. The annotation of rare and extreme phenotypic responses might be more informative than repeated adding of samples to an already well-annotated class, yet the user might not have the expertise in identifying the best training sample sets.

This limitation is addressed by active learning methods. The learning algorithm selects data points autonomously and presents them to the human expert for labeling. Data points are selected by the learning algorithm in order to maximize the learning progress, and hence, minimize the overall annotation effort (Jones et al., 2009). The criteria for selecting and proposing objects for annotation is typically based on uncertainty measures, whereby the most uncertain objects (from the perspective of the learner) are selected first. Similarly, interactive learning aims to shorten the feedback loop in the annotation process. Directly applying the learning result to other yet-unlabeled data samples allows the expert to inspect the current power of the learner visually and thus helps to identify cases with wrong predictions.

The prioritization of computer-selected data points can indeed improve the learning rate (Tomanek and Olsson, 2009) by guiding the human expert in establishing a comprehensive training data set (Fig. 4). Interactive learning requires fast algorithms and efficient software implementations and thus might not always be applicable.

Some experimental design guidelines

Reproducibility of the image-recording procedure is of utmost importance for the successful application of machine learning. Machine learning is designed to generalize from examples, but it will only generalize from variability that was present in the training data. For example, slight changes in the image focal plane, which might not even be noticed by a human observer, can introduce variability into the data that leads to systematic misclassifications. It is therefore strongly advisable to use autofocus devices to maximize reproducibility of image recording.

Similarly, the illumination intensity should be kept absolutely constant. Variable illumination intensities result in different noise levels, which can bias the classification. Conventional mercury or xenon light sources have variable illumination intensities depending on their lifetime and the heat-up time, for which compensation is required. New light sources, such as LEDs or solid-state lasers, yield a more stable output and are therefore preferable for machine-learning applications. Variable cell densities or differences in low-level image features owing to the experimental setup (such as microscope settings or different imaging media or incubation temperatures) that are not related to a biological phenotype can severely compromise the reliability of machine-learning methods (Shamir, 2011). An experimentalist should therefore keep environmental conditions as constant as possible. Data quality and reproducibility can be assessed by automated quality control (Zeder et al., 2010) and by incorporating control treatments in the assay. Differences in image features resulting from experimental variations are unlikely to be become obvious in the evaluation of the machine-learning method itself and thus have to be avoided early on in data acquisition and sample preparation.

Feature design has a great impact on the overall performance, as the learner can only learn what it has 'seen' in terms of features. The design and selection of optimal features can be difficult; however, general-purpose feature sets work well for most morphology-based assays (Hu and Murphy, 2004; Carpenter et al., 2006; Jones et al., 2008; Held et al., 2010). Engineering of specialized features might be necessary for specific biological assays, but should be envisioned only after unsuccessful application of general-purpose feature sets (Fig. 4).

Machine learning in cell biology – conclusions and outlook

Machine learning has tremendous power in the analysis of large-scale microscopic image data. Some representative examples for typical machine-learning applications are screens for mitotic regulators (Kittler et al., 2004; Neumann et al., 2010; Schmitz et al., 2010; Wurzenberger et al., 2012), control of cellular stress responses (Wippich et al., 2013), factors involved in ribosome biogenesis (Wild et al., 2010) and cellular host factors involved in virus infection (Mercer et al., 2012). Unsupervised machine learning has been used, for example, to study the heterogeneity of cell responses to diverse drugs (Loo et al., 2009; Singh et al., 2010), to construct genetic interaction profiles (Horn et al., 2011) and for automatic staging of mitotic progression (Zhong et al., 2012).

Current implementations of machine-learning software for cell biology have been optimized for the needs of large-scale screens. However, most cell biological studies are hypothesis driven and require frequent adaptations of the assay for testing small sets of candidate experimental perturbations. In such an experimental framework, many biologists still visually inspect data and develop quantification methods based on specific rule sets that are implemented manually as macros or software plug-ins. This approach is tedious and the data analysis often still requires some level of user interaction. By further improving the usability of software interfaces, machine learning could eventually replace most manually programmed analysis pipelines to facilitate assay development and increase processing throughput, accuracy and objectivity.

The power of machine learning can be further leveraged by a seamless integration into the image-acquisition process (Conrad et al., 2011). As state-of-the-art microscopes support full motorization and specimen interaction (e.g. by photobleaching at defined image areas or compound dispensing), automatic online recognition of phenotypes enables intelligent imaging workflows with highly sophisticated biological assays.

Funding

Research in the Gerlich laboratory has been supported by the European Community's Seventh Framework Programme (FP7/2007–2013) [grant numbers 241548 (MitoSys), 258068 (Systems Microscopy)]; a European Research Council Starting Grant [grant number 281198]; and the Austrian Science Fund (FWF)-funded project 'SFB Chromosome Dynamics'. Deposited in PMC for immediate release.

References

Ambroise, C. and McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA* **99**, 6562-6566.

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B. and Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLOS Comput. Biol.* **4**, e1000173.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Boland, M. V. and Murphy, R. F. (2001). A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics* **17**, 1213-1223.

Breiman, L. (2001). Random forests. *Mach. Learn.* **45**, 5-32.

Breiman, L., Friedman, J., Olshen, R., Stone, C., Steinberg, D. and Colla, P. (1983). *CART: Classification and Regression Trees*. Belmont, CA: Wadsworth.

Buchser, W., Collins, M., Garyantes, T., Guha, R., Haney, S., Lemmon, V., Li, Z. and Trask, O. J. (2004). Assay development guidelines for image-based high content screening, high content analysis and high content imaging. In *Assay Guidance Manual* (ed. G. S. Sittampalam, N. Gal-Edd, M. Arkin, D. Auld, C. Austin, B. Bejcek, M. Glicksman, J. Inglese, V. Lemmon, Z. Li et al.). Bethesda, MD: Eli Lilly & Company; National Center for Advancing Translational Sciences.

Buck, T. E., Li, J., Rohde, G. K. and Murphy, R. F. (2012). Toward the virtual cell: automated approaches to building models of subcellular organization "learned" from microscopy images. *Bioessays* **34**, 791-799.

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Guertin, D. A., 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.

Castelo, R. and Guigó, R. (2004). Splice site identification by idlBNs. *Bioinformatics* **20 Suppl. 1**, i69-i76.

Castoreno, A. B., Smurnyy, Y., Torres, A. D., Vokes, M. S., Jones, T. R., Carpenter, A. E. and Eggert, U. S. (2010). Small molecules discovered in a pathway screen target the Rho pathway in cytokinesis. *Nat. Chem. Biol.* **6**, 457-463.

Chebira, A., Barbotin, Y., Jackson, C., Merryman, T., Srinivasa, G., Murphy, R. F. and Kovacević, J. (2007). A multiresolution approach to automated classification of protein subcellular location images. *BMC Bioinformatics* **8**, 210.

Chen, Y. Q., Nixon, M. S. and Thomas, D. W. (1995). Statistical geometrical features for texture classification. *Pattern Recognit.* **28**, 537-552.

Collinet, C., Stöter, M., Bradshaw, C. R., Samusik, N., Rink, J. C., Kenski, D., Habermann, B., Buchholz, F., Henschel, R., Mueller, M. S. et al. (2010). Systems survey of endocytosis by multiparametric image analysis. *Nature* **464**, 243-249.

Conrad, C. and Gerlich, D. W. (2010). Automated microscopy for high-content RNAi screening. *J. Cell Biol.* **188**, 453-461.

Conrad, C., Erfle, H., Warnat, P., Daigle, N., Lörch, T., Ellenberg, J., Pepperkok, R. and Eils, R. (2004). Automatic identification of subcellular phenotypes on human cell arrays. *Genome Res.* **14**, 1130-1136.

Conrad, C., Wünsche, A., Tan, T. H., Bulkescher, J., Sieckmann, F., Verissimo, F., Edelstein, A., Walter, T., Liebel, U., Pepperkok, R. et al. (2011). Micropilot: automation of fluorescence microscopy-based imaging for systems biology. *Nat. Methods* **8**, 246-249.

Danuser, G. (2011). Computer vision in cell biology. *Cell* **147**, 973-978.

Datta, S. and Pihur, V. (2010). Feature selection and machine learning with mass spectrometry data. *Methods Mol. Biol.* **593**, 205-229.

de Ridder, D., de Ridder, J. and Reinders, M. J. (2013). Pattern recognition in bioinformatics. *Brief. Bioinform.* **14**, 633-647.

Devarajan, K. (2008). Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLOS Comput. Biol.* **4**, e1000029.

Doil, C., Mailand, N., Bekker-Jensen, S., Menard, P., Larsen, D. H., Pepperkok, R., Ellenberg, J., Panier, S., Durocher, D., Bartek, J. et al. (2009). RNF168 binds and amplifies ubiquitin conjugates on damaged chromosomes to allow accumulation of repair proteins. *Cell* **136**, 435-446.

Domingos, P. (2012). A few useful things to know about machine learning. *Commun. ACM* **55**, 78-87.

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.

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, pp. 23-37. New York, NY: Springer.

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.

Fuchs, F., Pau, G., Kranz, D., Sklyar, O., Budjan, C., Steinbrink, S., Horn, T., Pedal, A., Huber, W. and Boutros, M. (2010). Clustering phenotype populations by genome-wide RNAi and multiparametric imaging. *Mol. Syst. Biol.* **6**, 370.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J. et al. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.* **5**, R80.

Hamilton, N. A., Wang, J. T., Kerr, M. C. and Teasdale, R. D. (2009). Statistical and visual differentiation of subcellular imaging. *BMC Bioinformatics* **10**, 94.

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proc. IEEE* **67**, 786-804.

Hastie, T., Tibshirani, R., Friedman, J. and Franklin, J. (2005). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York, NY: Springer.

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.

Horn, T., Sandmann, T., Fischer, B., Axelsson, E., Huber, W. and Boutros, M. (2011). Mapping of signaling networks through synthetic genetic interaction analysis by RNAi. *Nat. Methods* **8**, 341-346.

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.

Hu, Y. and Murphy, R. F. (2004). Automated interpretation of subcellular patterns from immunofluorescence microscopy. *J. Immunol. Methods* **290**, 93-105.

Huang, K. and Murphy, R. F. (2004). Automated classification of subcellular patterns in multicell images without segmentation into single cells. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2004*, pp. 1139-1142. Chicago, IL: IEEE.

Huh, S., Lee, D. and Murphy, R. F. (2009). Efficient framework for automated classification of subcellular patterns in budding yeast. *Cytometry* **75A**, 934-940.

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. J., Grenier, J. K., Castoreno, A. B., Eggert, U. S., Root, D. E., Golland, P. et al. (2009). Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proc. Natl. Acad. Sci. USA* **106**, 1826-1831.

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.

Kanamori, T., Takenouchi, T., Eguchi, S. and Murata, N. (2007). Robust loss functions for boosting. *Neural Comput.* **19**, 2183-2244.

Kaynig, V., Fuchs, T. and Buhmann, J. M. (2010). Neuron geometry extraction by perceptual grouping in sstem images. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2902-2909. San Francisco, CA: IEEE.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing* **55**, 307-319.

Kittler, R., Putz, G., Pelletier, L., Poser, I., Heninger, A. K., Drechsel, D., Fischer, S., Konstantinova, I., Habermann, B., Grabner, H. et al. (2004). An endoribonuclease-prepared siRNA screen in human cells identifies genes essential for cell division. *Nature* **432**, 1036-1040.

Kittler, R., Pelletier, L., Heninger, A. K., Slabicki, M., Theis, M., Miroslaw, L., Poser, I., Lawo, S., Grabner, H., Kozak, K. et al. (2007). Genome-scale RNAi profiling of cell division in human tissue culture cells. *Nat. Cell Biol.* **9**, 1401-1412.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of The International Joint Conference on Artificial Intelligence*, Vol. 14, pp. 1137-1145. Montreal, QC: IJCAI.

Kotsiantis, S., Kanellopoulos, D. and Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering* **30**, 25-36.

Lansing Taylor, D., Haskins, J. R. and Giuliano, K. A. (2007). *High Content Screening*. Totowa, NJ: Humana Press.

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A. et al. (2006). Machine learning in bioinformatics. *Brief. Bioinform.* **7**, 86-112.

Lin, C., Hong, P., Bakal, C. and Perrimon, N. (2010). Novel morphological phenotypes discovery in high-content screens using underused features. In *Proceedings of the ISCA 2nd International Conference on Bioinformatics and Computational Biology*, pp. 184-189. Honolulu, HI: BICoB.

Lindblad, J., Wahlby, C., Bengtsson, E. and Zaltsman, A. (2004). Image analysis for automatic segmentation of cytoplasms and classification of Rac1 activation. *Cytometry* **57A**, 22-33.

Liu, S., Mundra, P. A. and Rajapakse, J. C. (2011). Features for cells and nuclei classification. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* pp. 6601-6604. Boston, MA: IEEE.

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

Lock, J. G. and Strömblad, S. (2010). Systems microscopy: an emerging strategy for the life sciences. *Exp. Cell Res.* **316**, 1438-1444.

Long, F., Peng, H., Liu, X., Kim, S. K. and Myers, E. (2009). A 3D digital atlas of C. elegans and its application to single-cell analyses. *Nat. Methods* **6**, 667-672.

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.

Loo, L. H., Lin, H. J., Singh, D. K., Lyons, K. M., Altschuler, S. J. and Wu, L. F. (2009). Heterogeneity in the physiological states and pharmacological responses of differentiating 3T3-L1 preadipocytes. *J. Cell Biol.* **187**, 375-384.

Mercer, J., Snijder, B., Sacher, R., Burkard, C., Bleck, C. K., Stahlberg, H., Pelkmans, L. and Helenius, A. (2012). RNAi screening reveals proteasome- and Cullin3-dependent stages in vaccinia virus infection. *Cell Reports* **2**, 1036-1047.

Meyer, D., Leisch, F. and Hornik, K. (2003). The support vector machine under test. *Neurocomputing* **55**, 169-186.

Millard, B. L., Niepel, M., Menden, M. P., Muhlich, J. L. and Sorger, P. K. (2011). Adaptive informatics for multifactorial and high-content biological data. *Nat. Methods* **8**, 487-492.

Mirabeau, O., Perlas, E., Severini, C., Audero, E., Gascuel, O., Possenti, R., Birney, E., Rosenthal, N. and Gross, C. (2007). Identification of novel peptide hormones in the human proteome by hidden Markov model screening. *Genome Res.* **17**, 320-327.

Murphy, R. F. (2011). An active role for machine learning in drug development. *Nat. Chem. Biol.* **7**, 327-330.

Myers, G. (2012). Why bioimage informatics matters. *Nat. Methods* **9**, 659-660.

Neumann, B., Walter, T., Hériché, J. K., Bulkescher, J., Erfle, H., Conrad, C., Rogers, P., Poser, I., Held, M., Liebel, U. et al. (2010). Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature* **464**, 721-727.

Ng, A. and Jordan, A. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process. Syst.* **14**, 841-848.

Oliveira, F. P. M. and Tavares, J. M. R. S. (2012). Medical image registration: a review. *Comput. Methods Biomech. Biomed. Engin*. [Epub ahead of print] doi:10.1080/10255842.2012.670855.

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. (2013). imageHTS: Analysis of high-throughput microscopy-based screens.

Perlman, Z. E., Slack, M. D., Feng, Y., Mitchison, T. J., Wu, L. F. and Altschuler, S. J. (2004). Multidimensional drug profiling by automated microscopy. *Science* **306**, 1194-1198.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257-286.

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

Rajaram, S., Pavie, B., Hac, N. E., Altschuler, S. J. and Wu, L. F. (2012b). SimuCell: a flexible framework for creating synthetic microscopy images. *Nat. Methods* **9**, 634-635.

Reiter, L., Rinner, O., Picotti, P., Hüttenhain, R., Beck, M., Brusniak, M. Y., Hengartner, M. O. and Aebersold, R. (2011). mProphet: automated data processing and statistical validation for large-scale SRM experiments. *Nat. Methods* **8**, 430-435.

Saeys, Y., Inza, I. and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**, 2507-2517.

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.

Schmitz, M. H. A., Held, M., Janssens, V., Hutchins, J. R. A., Hudecz, O., Ivanova, E., Goris, J., Trinkle-Mulcahy, L., Lamond, A. I., Poser, I. et al. (2010). Live-cell imaging RNAi screen identifies PP2A-B55alpha and importin-beta1 as key mitotic exit regulators in human cells. *Nat. Cell Biol.* **12**, 886-893.

Shamir, L. (2011). Assessing the efficacy of low-level image content descriptors for computer-based fluorescence microscopy image analysis. *J. Microsc.* **243**, 284-292.

Shamir, L., Orlov, N., Eckley, D. M., Macura, T., Johnston, J. and Goldberg, I. G. (2008). Wndchrm - an open source utility for biological image analysis. *Source Code Biol. Med.* **3**, 13.

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.

Shariff, A., Kangas, J., Coelho, L. P., Quinn, S. and Murphy, R. F. (2010). Automated image analysis for high-content screening and analysis. *J. Biomol. Screen.* **15**, 726-734.

Singh, D. K., Ku, C. J., Wichaidit, C., Steininger, R. J., 3rd, Wu, L. F. and Altschuler, S. J. (2010). Patterns of basal signaling heterogeneity can distinguish cellular populations with different drug sensitivities. *Mol. Syst. Biol.* **6**, 369.

Slack, M. D., Martinez, E. D., Wu, L. F. and Altschuler, S. J. (2008). Characterizing heterogeneous cellular responses to perturbations. *Proc. Natl. Acad. Sci. USA* **105**, 19306-19311.

Sommer, C., Straehle, C., Kothe, U. and Hamprecht, F. A. (2011). Ilastik: interactive learning and segmentation toolkit. In *Proceedings of the 2011 8th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 230-233. Chicago, IL: IEEE.

Sommer, C., Held, M., Fischer, B., Huber, W. and Gerlich, D. W. (2013). CellH5: a format for data exchange in high-content screening. *Bioinformatics* **29**, 1580-1582.

Swedlow, J. R. and Eliceiri, K. W. (2009). Open source bioimage informatics for cell biology. *Trends Cell Biol.* **19**, 656-660.

Tarca, A. L., Carey, V. J., Chen, X. W., Romero, R. and Drăghici, S. (2007). Machine learning and its applications to biology. *PLOS Comput. Biol.* **3**, e116.

Thévenaz, P., Ruttimann, U. E. and Unser, M. (1998). A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Process.* **7**, 27-41.

Tomanek, K. and Olsson, F. (2009). A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pp. 45-48. Boulder, CO: NAACL.

Tu, Z. and Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3D brain image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 1744-1757.

Van der Maaten, L., Postma, E. and Van Den Herik, H. (2009). Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.* **10**, 1-41.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. New York, NY: Springer.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int. J. Comput. Vis.* **57**, 137-154.

Wang, J., Zhou, X., Bradley, P. L., Chang, S. F., Perrimon, N. and Wong, S. T. (2008). Cellular phenotype recognition for high-content RNA interference genome-wide screening. *J. Biomol. Screen.* **13**, 29-39.

Weber, S., Fernández-Cachón, M. L., Nascimento, J. M., Knauer, S., Offermann, B., Murphy, R. F., Boerries, M. and Busch, H. (2013). Label-free detection of neuronal differentiation in cell populations using high-throughput live-cell imaging of PC12 cells. *PLoS ONE* **8**, e56690.

Wild, T., Horvath, P., Wyler, E., Widmann, B., Badertscher, L., Zemp, I., Kozak, K., Csucs, G., Lund, E. and Kutay, U. (2010). A protein inventory of human ribosome biogenesis reveals an essential function of exportin 5 in 60S subunit export. *PLoS Biol.* **8**, e1000522.

Wippich, F., Bodenmiller, B., Trajkovska, M. G., Wanka, S., Aebersold, R. and Pelkmans, L. (2013). Dual specificity kinase DYRK3 couples stress granule condensation/dissolution to mTORC1 signaling. *Cell* **152**, 791-805.

Wurzenberger, C., Held, M., Lampson, M. A., Poser, I., Hyman, A. A. and Gerlich, D. W. (2012). Sds22 and Repo-Man stabilize chromosome segregation by counter-acting Aurora B on anaphase kinetochores. *J. Cell Biol.* **198**, 173-183.

Yang, Z. R. and Chou, K. C. (2004). Bio-support vector machines for computational proteomics. *Bioinformatics* **20**, 735-741.

Zeder, M., Kohler, E. and Pernthaler, J. (2010). Automated quality assessment of autonomously acquired microscopic images of fluorescently stained bacteria. *Cytometry A* **77**, 76-85.

Zhong, Q., Busetto, A. G., Fededa, J. P., Buhmann, J. M. and Gerlich, D. W. (2012). Unsupervised modeling of cell morphology dynamics for time-lapse microscopy. *Nat. Methods* **9**, 711-713.