Chapter 7

Quality Control for High-Throughput Imaging Experiments Using Machine Learning in Cellprofiler

Mark-Anthony Bray and Anne E. Carpenter

Abstract

Robust high-content screening of visual cellular phenotypes has been enabled by automated microscopy and quantitative image analysis. The identification and removal of common image-based aberrations is critical to the screening workflow. Out-of-focus images, debris, and auto-fluorescing samples can cause artifacts such as focus blur and image saturation, contaminating downstream analysis and impairing identification of subtle phenotypes. Here, we describe an automated quality control protocol implemented in validated open-source software, leveraging the suite of image-based measurements generated by CellProfiler and the machine-learning functionality of CellProfiler Analyst.

Key words Cell-based assays, High-content screening, Image analysis, Microscopy, Quality control, Machine learning, Open-source software

1 Introduction

The use of automated microscopy combined with image analysis methods has enabled the extraction of quantitative image-based information from cells, tissues, and organisms while speeding analysis and reducing subjectivity (*see* refs. 1, 2). Any number of high-content assays can be quantified by combining high-resolution microscopy with sophisticated image analysis techniques in order to create an automated workflow with a high degree of reproducibility, fidelity, and robustness (*see* ref. 3). Analyzing experiments that are comprised of tens to millions of images allows for quantitative modeling of biological processes and discerning complex and subtle phenotypes.

However, reliable downstream processing of such datasets often depends on robust exclusion of images that would otherwise be erroneously scored as screening hits or inadvertently ignored as false negatives. Abnormalities in image quality can degrade otherwise high-quality microscopy data and, in severe cases, even render

Paul A. Johnston and Oscar J. Trask (eds.), *High Content Screening: A Powerful Approach to Systems Cell Biology and Phenotypic Drug Discovery*, Methods in Molecular Biology, vol. 1683, DOI 10.1007/978-1-4939-7357-6_7,

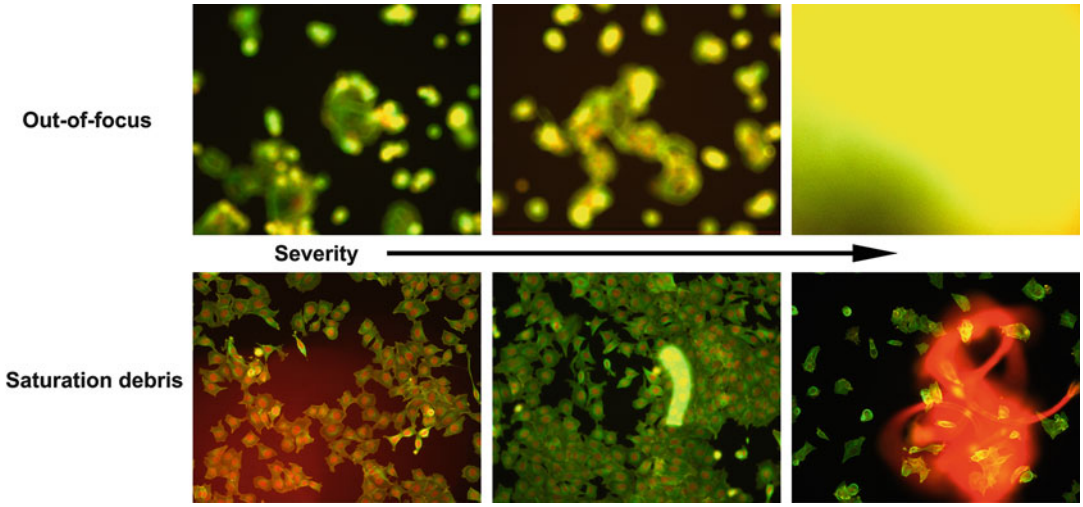

Fig. 1 Examples of HCS images containing artifacts. Out-of-focus (*top row*) and saturation debris (*bottom row*) examples are shown. Images are taken from the Broad Bioimage Benchmark Collection (BBBC) at http:/www.broadinstitute.org/bbbc/BBBC021/. These images come from a compound mechanism of action assay consisting of MCF-7 cells labeled with fluorescent markers for DNA (*red*), β-tubulin (*green*), and actin filaments (*yellow*)

some experimental approaches infeasible. In our experience, as many as 5% of the fields of view in a routine screen can be affected with such artifacts to varying degrees. For high-throughput assays, manual inspection of all images for quality control (QC) purposes is not tractable; therefore, the development of QC methodologies must be similarly automated to keep up with the increasing demands of modern imaging experiments.

This chapter outlines a protocol for the characterization of images for common artifacts that confound high-content imaging experiments, including focus blur and image saturation (Fig. 1). The protocol uses the open-source, freely downloadable software packages, CellProfiler and CellProfiler Analyst. CellProfiler has been validated for a diverse array of biological applications, typically for generating features on a per-cell basis (*see* refs. 4, 5). Likewise, CellProfiler Analyst has been previously used for per-cell classification of phenotypes (*see* refs. 4, 6). The workflow described below expands our prior work using CellProfiler and CellProfiler Analyst validating image-based metrics for QC (*see* ref. 7) and provides a step-by-step protocol that leverages the functionality of both of these packages for QC purposes.

2 Materials

2.1 High-Content Fluorescent Images for Assessment

1. Either single channel or multichannel fluorescent images acquired on a microscopy platform, conventionalor automated, may be analyzed. CellProfiler is capable of handling both

fluorescence and transmitted-light (e.g., bright-field) images; however, this protocol assumes that a fluorescent assay is being evaluated (*see* **Note 1**).

2. More than 120 file formats are readable by CellProfiler, including TIF, BMP, and PNG; standardized HCS image data formats such as OME-TIFF are also supported. Some file formats are more amenable to image analysis than others (*see* **Note 2**).
3. For most screening applications, images are captured by an automated microscope from multi-well plates, such that each image is annotated with unique plate, well, and site metadata identifiers. Using this metadata will enable some features in CellProfiler Analyst, as described below.
4. The images may be contained in a single folder or in a set of folders or subfolders. While hundreds of images may be analyzed on a single computer, such a computing solution is insufficient for the thousands or millions of images characteristic of large-scale screens. In the latter case, a CellProfiler analysis can be run on a computing cluster, taking advantage of the hardware infrastructure to process any number of images in parallel (*see* Subheading 8.1).
5. An example screening image set (BBBC021v1) is available from the Broad Bioimage Benchmark Collection (BBBC) at http://www.broadinstitute.org/bbbc/BBBC021/ (*see* refs. 8, 9). These images come from a small molecule mechanism of action assay consisting of MCF-7 cells labeled with fluorescent markers for DNA, β-tubulin, plasma membrane, and actin filaments.

2.2 A Desktop or Laptop Computer

1. A Mac, PC, or Linux computer with at least 4 GB of RAM, a 2 GHz processor and a 64-bit processor is recommended. If the images are stored remotely, a fast Internet connection is recommended for rapid image loading.
2. A single image set such as those in the example set demonstrated here will be processed in <1 min/image on a single computer with a 2.67 GHz processor and 8 GB RAM.
3. Large image sets (greater than ~1000 images) will likely require a computing cluster (*see* **Note 3**).

2.3 CellProfiler and CellProfiler Analyst Software

1. Both applications are free and open-source (BSD license).
2. The CellProfiler image analysis software package is available as a distributable installation package for Windows and Mac and can be downloaded at http://cellprofiler.org/. This protocol uses CellProfiler version 2.2.0. Researchers who wish to implement their own image analysis algorithms or run CellProfiler on UNIX/Linux or a computing cluster will want to download

the source code (*see* **Note 4**). All versions are free and open-source (BSD license).

3. The CellProfiler Analyst software package is available for Windows and Mac as a distributable installer package at http://cellprofiler.org/. This protocol uses CellProfiler version 2.2.0.
4. For both packages, follow the installation instructions from their respective download pages. If difficulties on this step are encountered, visit the online forum (http://forum.cellprofiler.org/) to search if the problem has been previously encountered and resolved, or post the issue to the forum.

3 Methods

The protocol begins with configuring the input and output file locations for the CellProfiler program and constructing a modular QC "pipeline". Image processing modules are selected and placed in the pipeline and the modules' settings are adjusted appropriately according to the specifics of the HCS project (for example, spatial scales for blur measurements, and the channels used for thresholding; see the section "Configuring the `MeasureImageQuality` module" below). The pipeline is then run on the images collected in the experiment to assemble a suite of QC measurements, including the image's power log-log slope, textural correlation, percentage of the image occupied by saturated pixels, and the standard deviation of the pixel intensities, among others. These measurements are used within the machine-learning tool packaged with CellProfiler Analyst to automatically classify images as passing or failing QC criteria determined by a classification algorithm. The results can either be written to a database for further review, or the classifier can be used to filter images within a later CellProfiler pipeline so that only those images which pass QC are used for cellular feature extraction. An overview of the workflow is shown in Fig. 2.

3.1 Starting CellProfiler and Loading a Pipeline

1. Start CellProfiler by selecting CellProfiler from the Start Menu (Windows) or Applications folder (Mac), or from the command line (any OS). The CellProfiler welcome screen and graphical user interface will appear (Fig. 3a).
2. Download an example quality control (QC) pipeline from http://pubs.broadinstitute.org/bray_methodsmolbiol_2016/. From the main menu bar, select *File* > *Import* > *Pipeline from File...* and browse to the location of the downloaded pipeline (or alternately, drag/drop the pipeline file into the CellProfiler pipeline panel). This will load the QC pipeline which can be adjusted as needed for other assays; the details on the specific settings are described in the following sections and associated notes.

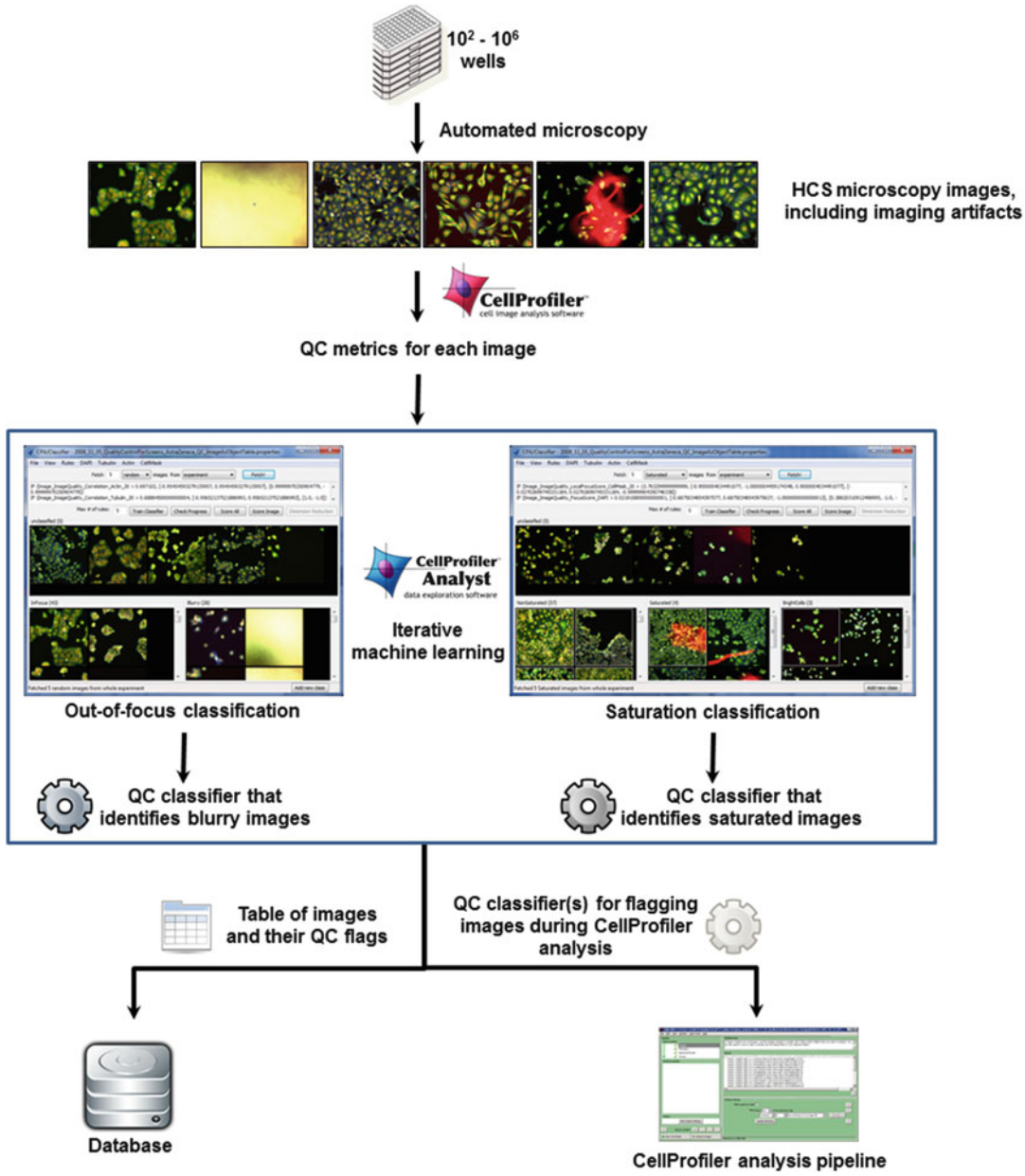

Fig. 2 Overall quality control workflow. A suite of image quality measures are obtained by CellProfiler from images collected by an automated microscope. These measurements are used as input into a supervised machine learning tool in CellProfiler Analyst; the researcher then trains the computer to classify images as out-of-focus or containing saturation artifacts. The classifier then scores all images from the experiment, with the QC results stored as metadata in a database, or the classifier incorporated into an analysis pipeline

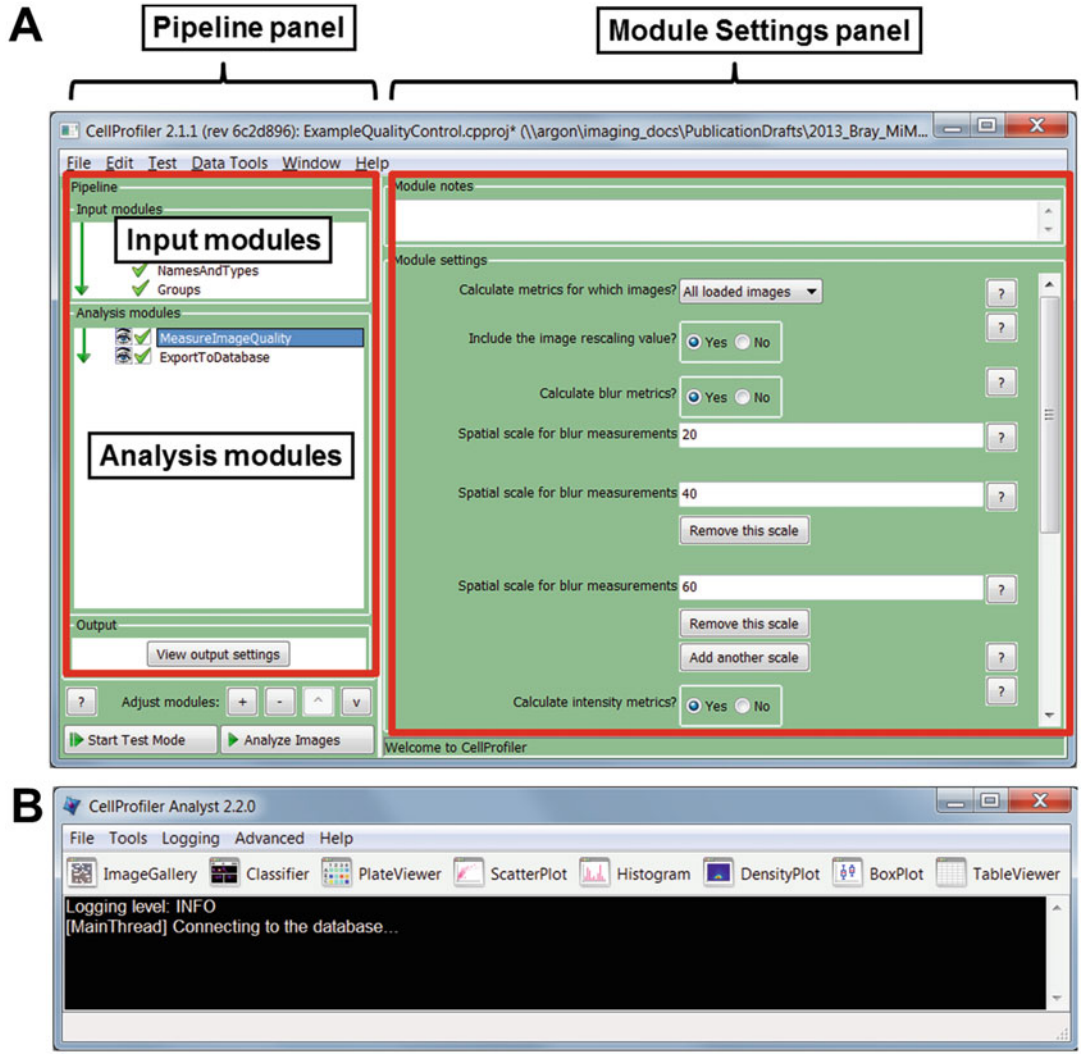

Fig. 3 Screenshots of the software packages described in the protocol. (**a**) The CellProfiler interface. The Pipeline panel is divided into three sections: the Input modules which specify information about the images to be processed, the Analysis modules which are executed sequentially to collect the measurements, and the Output, specifying the location of the output files. The Module settings panel provides the customizable settings for each selected module in the Pipeline panel. (**b**) The CellProfiler Analyst user interface. The Classifier tool (*icon on the upper left*) is used to train a classifier to distinguish between images of various types; other icons launch tools used for data visualization and exploration

3. Because CellProfiler is usable on a wide variety of assays, only the modules and associated settings relevant to the quality-control protocol are listed and described in this protocol. The remaining module settings are not mentioned and should be adjusted to suit each specific assay set as needed; each module

has extensive documentation to assist with fine-tuning the settings (*see* **Note 5**). Other modules can also be added and positioned if further image measurements are desired (*see* **Note 6**).

3.2 Configuring Image Input for CellProfiler

1. Select the `Images` module, the first module in the Input modules section of the pipeline panel. A box will appear in the module settings panel prompting for files to be placed into the box. Drag-and-drop the desired image files (or folders containing the image files), as described in the Materials section, into this box; the box will update and show a listing of the collected files. These files will be used as input into the QC pipeline. Adding entire folders of files is acceptable even if some of the contents are not to be processed; these can be filtered in the next step.
2. Adjust the "Filter images?" drop-down below the file list box to specify what files in the file list are passed downstream for further processing. The default setting is "Images only," which is sufficient for most data sets. If only a subset of files are to be used as input (e.g., only process those files with the extension "TIF"), select "Custom" from the drop-down box and then define rules for filtering the files for processing (*see* **Note 7**).

3.3 Specifying Image Metadata (Optional)

1. If there is information (metadata) that is associated with the images, such as experiment, plate, and well identities, select the `Metadata` module (the second module of the Input modules) and select for "Yes" for "Extract metadata?" This module should definitely be used if information about the well layout is contained in the image filename or folder name. Configure the module according to the settings listed below.
2. *Metadata extraction method*: Select "Extract from file/folder names" if the metadata information is contained within the image filename or path, then select "File name" or "Folder name" from the *Metadata source* setting that appears. Select "Import metadata" if it is contained in a comma-delimited file (CSV) of values, then browse to the file location from the setting that appears.
3. *Regular expression*: This setting may require adjustment to match the nomenclature applied by the acquisition software (*see* **Note 8**). However, the default of "^(?P<Plate>.*)_(?P<Well>[A-P][0–9]{2})_s(?P<Site>[0–9])_w(?P<ChannelNumber>[0–9])" is sufficient for a number of commercial systems (*see* **Note 9**).
4. If additional metadata needs to be included, click the "Add another extraction method" button to reveal additional

settings which can then be adjusted to include further metadata sources such as sample treatment information.

5. Click the "Update" button below the horizontal divider to display a table where each row shows an input image's filename and whatever associated metadata is available, including plate layout identifiers such as plate, well, and site, as well as sample and treatment information.

3.4 Specifying the CellProfiler Name and Type of Image Channels

1. Select the `NamesAndTypes` module, the third module in the Input modules section of the pipeline panel. This module is used to assign a user-defined name to particular images or channel(s), and define their relationship to one another. Configure the module according to the settings listed below.
2. *Assign a name to*: Select "Images matching rules" to select a subset of images from the `Images` module as belonging to the same channel.
3. *Select the rule criteria*: From the drop-down and edit boxes, select an identifier and the value for this identifier in order to distinguish a subset of images as a unique channel. In the example pipeline, the settings are specified as: "Metadata", "Does", and "Have ChannelNumber matching" in the three drop-down menus, and "1" is entered in the edit box. This combination of settings will identify those images which have the ChannelNumber metadata identifier specified as "1" and ignore all others. If no metadata was gathered from the `Metadata` module, then other image characteristics such as filename, extension, and image type may be used to identify a unique channel.
4. *Name to assign these images*: Enter a suitably descriptive name to identify the image for later use in the pipeline; downstream modules will then refer to the image by this name for processing. For example, "DNA" can be used to indicate that the first wavelength corresponds to DNA-stained images.
5. *Select the image type*: Select the image format that corresponds to this channel (*see* **Note 10**).
6. Press the "Add another image" button if the assay involves multiple channels; additional settings will be revealed so that further matching rules and names can be given to additional channels. Any number of channels may be specified using this method.
7. *Image set matching method*: This step associates multiple image channels with each other, for each field of view. If the `Metadata` module was used to specify the identifiers for the channel, select "Metadata" to display a panel containing a column for each channel given above, and a row of drop-down menus with available metadata identifiers. For each row, match the

metadata identifiers so that the channels are properly matched together (*see* **Note 11**).

8. Press the "Update" button below the horizontal divider to display a table where each row displays a unique metadata combination, and the image names are listed as columns. When the pipeline executes during the analysis run, each set of images specified in a row will be loaded and processed as an individual image set. Check the listing for any errors, e.g., image channel mismatches.

3.5 Configuring the `MeasureImageQuality` Module

1. Select the `MeasureImageQuality` module, located in the Analysis modules panel below the Input modules. This module measures features that indicate image quality, including measurements of blur (poor focus), intensity, and saturation. Configure the module according to the settings listed below.
2. *Calculate metrics for which images?* Select "All loaded images" in order to calculate QC metrics for all channels that were specified in the `NamesAndTypes` module. Choose "Select..." to select a subset of these channels.
3. *Calculate blur metrics?* Select "Yes" for this setting to calculate a set of focus blur metrics, one for each channel specified above (*see* **Note 12**).
4. *Spatial scale for blur measurements*: Enter a number specifying the size(s) of the relevant features, in pixels. For a given amount of focus blur, the degradation of image quality will depend in part on the size of the cellular features imaged. For example, nuclei that are typically 20 pixels in diameter may not be as affected by a small amount of blurring as thin actin filaments that are only 5 pixels wide. Since the size of the features can vary over a wide range in HCS, it is often helpful to specify several spatial scales in order to capture differing amounts of blur; 0.5×, 1×, and 2× the size of a given structure of interest are good starting points. CellProfiler will measure the blurriness metrics for all specified spatial scales, for all selected input images; click the "Add another scale" button to include additional spatial scales. Later in the analysis, blur measurements resulting from each scale can be examined and assessed for utility.
5. *Calculate intensity metrics?* Select "Yes" for this setting to calculate a set of intensity measurements, one for each channel specified above (*see* **Note 13**).
6. *Calculate saturation metrics?* Select "Yes" for this setting to calculate a set of saturation metrics, one for each channel specified above (*see* **Note 13**).

3.6 Configuring the `ExportToDatabase` Module

1. Select the `ExportToDatabase` module in the Analysis modules panel. This module exports measurements produced by a pipeline directly to a database, which will be accessed by CellProfiler Analyst. Configure the module according to the settings listed below.
2. *Database type*: Select "MySQL" if remote access to a MySQL database is available; enter the host, username, and password information in the settings below. Select "SQLite" to instead store the data on hard drive space, such as a personal computer or networked storage space; this option does not require setting up or configuring a database (*see* **Note 14**).
3. *Experiment name*: Enter a name for the experiment; this can be any descriptor that uniquely identifies the analysis run. This name will be registered in the database and linked to the tables that `ExportToDatabase` creates.
4. *Database name*: Enter the name of the database that will store the collected measurements. CellProfiler will create the table if it does not exist already, or produce a warning prior to overwriting an existing table. Often this will be the same as *Table prefix* (*see* below).
5. If using "MySQL" for the database type, press the "Test connection" button to confirm the connection settings entered above.
6. *Overwrite without warning*? This setting will determine whether the database tables used to store the measurements will be created based on the researcher's response to a prompt at the beginning of the analysis run ("Never"), existing tables will be reused and measurements added or overwritten as needed ("Data only"), or the tables will be created without prompting ("Data and schema"). Be very careful with this setting, as it will enable overwriting existing data with the same database name.
7. *Add a prefix to table names*? Select "Yes" to this setting to uniquely specify the names of the tables created in the analysis run. If so, a setting labeled *Table prefix* will appear for entering the chosen identifier for the analysis run. This text will be prepended onto the default table name of "Per_Image" created in the database specified above; using a unique identifier allows multiple data tables to be written to the same database rather than over-writing the default table name with each run. For example, a QC pipeline run on the BBBC images described above (*see* Subheading 2) could use the prefix "BBBC021_QC" to distinguish the database table from QC runs performed on other BBBC images.
8. *Create a CellProfiler Analyst file*? Select "Yes" to this setting to create a configuration file (the "properties" file, described in

more detail in Subheading 3.9) that will be used by CellProfiler Analyst to access the images and measurements. Additional settings will appear upon selecting this option and are specified below.

9. *Access CPA images via URL?* Select "Yes" to this setting if the images are stored remotely and can be accessed via HTTP. If so, a setting labeled *Enter an image url prepend if you plan to access your files via http* will appear for entering a URL prefix. This prefix will be prepended onto all image locations during the analysis run. For example, if this setting is given as "http://some_server.org/images" and the path and file name in the database for a given image are "some_path" and "file.png," respectively, then CellProfiler Analyst will open "http://some_server.org/images/some_path/file.png".
10. *Select the plate type*: If using a multi-well plate assay, select the plate format from the drop-down box. Permissible types are 6, 24, 96, 384, 1536, and 5600 (for certain cell microarrays).
11. *Select the plate metadata*: If using multi-well plates, select the metadata identifier corresponding to the physical plate ID, otherwise leave as "None".
12. *Select the well metadata*: If using multi-well plates, select the metadata identifier corresponding to the well ID of the physical plates, otherwise leave as "None".
13. *Select the classification type*: Choose "Image" for this setting to enable image-based classification.
14. *Calculate the per-image mean values of object measurements?* Select "No" for this setting, because no objects are identified or measured in this pipeline.
15. *Export measurements for all objects to the database?* Select "None" from the drop-down box, because no objects are identified or measured in this pipeline. Note that a red triangle indicating module error on the setting "*Which objects should be used for locations?*" will disappear once this selection is made.
16. *Export object relationships?* Select "No" to this setting, because no objects are identified or measured in this pipeline.
17. *Write image thumbnails directly to the database?* Select "Yes" to this setting to write a miniature version of each image to the database. This is not necessary for the protocol described here, but may be helpful if using the PlateViewer tool in CellProfiler Analyst to explore images in a multi-well format.
18. *Select the images for which you want to save thumbnails*: Select the channels to be saved as thumbnails; use Ctrl-Click (Windows) or Command-Click (Mac) to select multiple channels.

3.7 Configuring the CellProfiler Output Settings

1. Click the button "View output settings" located at the bottom of the pipeline panel.
2. In the module settings panel, set the *Default Output Folder* to a folder that will contain the output. It is best to avoid spaces or special characters in naming the output folder. If this folder does not exist, it should be created beforehand using a file manager tool (e.g., Windows Explorer, Mac Finder), or by clicking the "New folder" button to the right of the Default Output Folder edit box.
3. Disable the creation of alternative-format output files by selecting the "Do not write MATLAB or HDF5 files" from the *Output file format* drop-down box; this additional output is not needed. Note that the regular-format output will still be produced by the `ExportToDatabase` module in the pipeline; this will be used for QC purposes.

3.8 Running the QC Pipeline

1. If running on a computing cluster, the `CreateBatchFiles` module must be added to the pipeline (*see* **Note 15**).
2. From the main menu bar, select *Window* > *Hide all windows on run*. By default, a window displaying the module results is typically opened for each module during the analysis run. Disabling these windows is recommended for the analysis run, as they will unnecessarily add to the overall run-time, and for the QC pipeline are rather uninformative.
3. Click the "Analyze images" button to begin the analysis processing run.
4. Upon starting the analysis, each image (or collection of images if multiple wavelengths are available) is processed by each module in the pipeline, in order.

3.9 Starting CellProfiler Analyst

1. Start CellProfiler Analyst (CPA) as instructed in the installation help.
2. A dialog box will appear requesting a "properties file." This file was produced by the CellProfiler QC pipeline and has the extension ".properties"; it is placed into the Default Output Folder specified by CellProfiler. This file contains the location of the database containing the QC measurements, image locations, and other associated information. Browse to the location of the properties file created above.
3. Once the properties file is loaded, the CPA interface will then appear (Fig. 3b). CellProfiler Analyst provides an interface with icons to launch a variety of tools.

3.10 Using the Classifier Tool to Detect Blurred Images

1. The exploration tools in CPA (e.g., PlateViewer, ScatterPlot and Histogram) are recommended for use if evaluation of a single QC measurement is sufficient to pass or fail an image (*see*

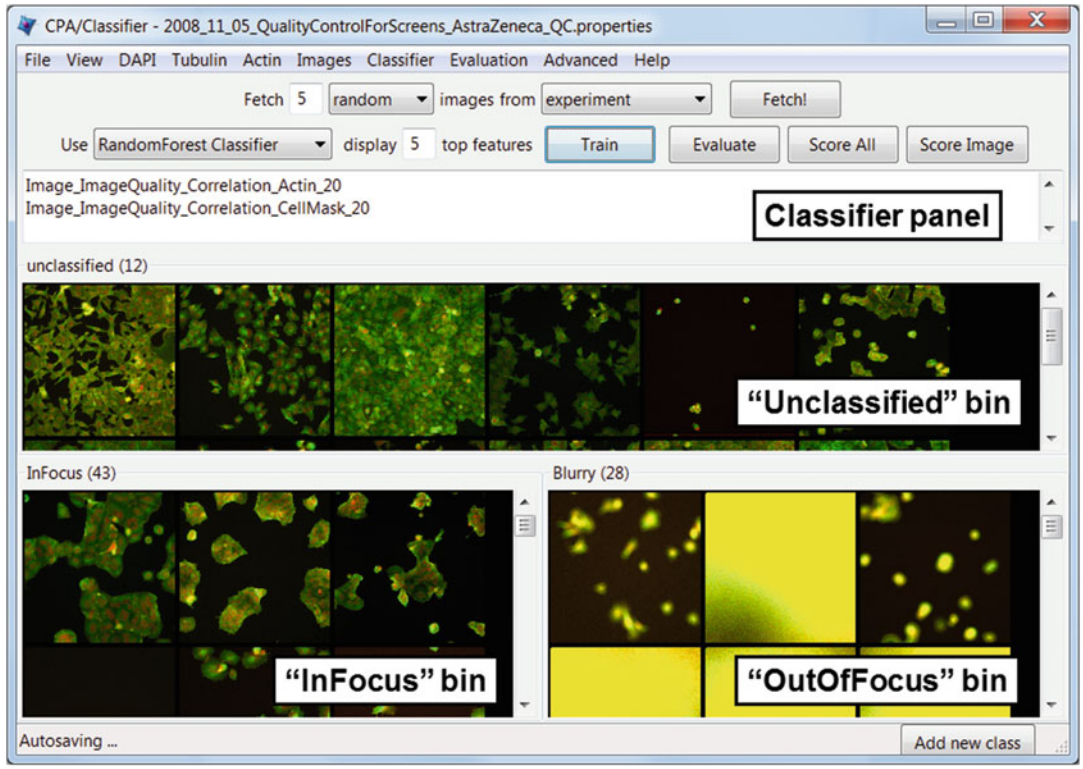

Fig. 4 Example screenshot of the Classifier tool. The panel for adjusting the type and number of images to retrieve ("fetch") is at the *top*. The classifier panel contains the top-scoring QC image features that Classifier has determined are best to distinguish the images in different bins. The "Unclassified" bin contains images that have yet to be sorted into a classification bin. The "InFocus" and "OutOfFocus" bins contain the training set images, that is, images designated by the user as belonging to one of the two classes; these samples will be used to generate the classifier sufficient to distinguish the classes

Note 16). However, if this is not the case, the following steps describe how machine-learning methods may be applied in order to automatically discern which measurements and cutoffs best apply to detect a given type of QC problem (a given QC "class").

2. Click the Classifier icon in the CPA interface to start the machine-learning tool Classifier (Fig. 4). Classifier trains the computer to discriminate user-defined image classes by iteratively applying a supervised machine-learning approach to the CellProfiler-generated QC measurements.
3. Right-click inside one of the bins located at the bottom to display a popup menu of options. Select the "Rename class" option and rename the bin to "InFocus"; this bin will contain examples of images that, by visual inspection, are properly focused. Rename the other bin to "OutOfFocus"; it will hold examples that fail QC due to blurriness.

4. In the top portion of the Classifier window, enter the number of images Classifier should retrieve (or "fetch"). The default of 20 images is a good starting point.
5. Click the "Fetch" button. Images will be randomly selected from the entire image set, and image tiles will begin to appear in the "Unclassified" bin (*see* **Note 17**).
6. Use the mouse to drag and drop the unclassified images into one of the two classification bins, "InFocus" or "OutOfFocus" (Fig. 4). Continue fetching and sorting images until at least ten examples populate each bin (*see* **Note 18**). If no examples of aberrant images are fetched after checking a few dozen, use the exploration tools to assist in finding a few examples based on some of the QC metrics measured by CellProfiler (*see* **Note 16**). The collection of images that have been annotated by classifying them (and are thus located in the lower bins of Classifier) is referred to as the *training set* (*see* **Note 19**).
7. In the top portion of the Classifier window, select the desired classifier from the drop-down box. We recommend using the default settings as a starting point (*see* **Note 20**).
8. Click the "Train" button. Depending on the classifier selected, the large text field near the top of the CPA interface will then populate with a list of the most important features selected by the initial classifier based on the training set; the machine learning algorithm is attempting to differentiate between the samples in each bin based on a combination of QC metrics measured by CellProfiler.
9. In the fetch controls (top part of the window), select "OutOfFocus" from the left-most drop-down menu. Click the "Fetch" button: Classifier will select examples that it deems as out-of-focus based on the current classifier and display them in the Unclassified bin. If the blurriness to be deemed as aberrant is fairly subtle, it may be helpful to fetch from the "InFocus" class images to make sure that only normal images are returned.
10. Correct any misclassifications you see (i.e., in-focus images classified as "OutOfFocus") by sorting them into the appropriate bins.
11. Click the "Train" button to revise the classifier based on the updated training set.
12. Repeat the above process of fetching images, sorting them into their appropriate classes, and re-training to improve the classifier until the results are sufficiently accurate (*see* **Note 21**). Two approaches for checking the classifier accuracy are provided under the "Evaluation" menu item: a confusion matrix displaying the fraction of images falling into the actual versus predicted class or a classification report displaying the precision,

recall, and F1-score for each class (*see* **Note 22**). Once an evaluation selection has been made, press the "Evaluate" button to generate the statistics, and *see* if more training samples are needed.

3.11 Using the Classifier Tool to Detect Saturated Images

1. Open another Classifier tool from the CPA main interface.
2. Right-click inside one of the bins located at the bottom to displays a popup menu of options. Rename the bin to "Non-Saturated". Rename the other bin to "Saturated".
3. Add additional bins if more classes are needed to discern subtleties between artifacts (*see* **Note 23**). Press the "Add new class" button at the bottom-right corner of the Classifier tool. At the prompt, give the new bin a descriptive name and a third bin will appear next to the others.
4. However many bins are needed to distinguish the desired classes, proceed with same procedure as in Subheadings 10.4–10.12 above, identifying images with various types of saturation (bright debris, whole-well fluorescence, etc.).

3.12 Saving the QC Results for Later Use

1. Save the training sets (for both the saturated and out-of-focus classifications) for future refinement, to regenerate a classifier across CPA sessions (but *see* **step 2**), and as an experimental record by selecting *File* > *Save Training Set* from the menu bar. It is advisable to do so periodically during the creation of a training set, but certainly before proceeding to scoring the experiment because scoring may take a long time for large screens.
2. Likewise, save the classifier generated by CPA (for both the saturated and out-of-focus classifications) by selecting *File* > *Save Classifier Model* (*see* **Note 24**). This classifier may also be used as part of a downstream CellProfiler analysis workflow; *see* Subheading 13 for details.
3. Click the "Score All" button to have Classifier score all images in the entire experiment. A dialog box will appear with scoring options; use the defaults of "Image" under "Grouping," and "None" under "Filter". Make sure the "Report enrichments?" box is unchecked before pressing the OK button, because this is only relevant for classifying individual objects.
4. Once scoring is completed, the results are presented in a Table-Viewer. Saving this table to a comma-delimited file (CSV) or to the original database can be done via *File* > *Save table to CSV* or *File* > *Save table to database*, respectively. In the latter case, the table can either be stored permanently or for the current session which means the table will be removed from the database when CPA is closed.

3.13 Using the QC Classification Results for CellProfiler Analysis (Optional)

1. The QC workflow described above is intended to form the initial steps of a larger data analysis workflow (*see* refs. 7, 10). Our laboratory typically runs the QC workflow prior to completing a full analysis of images using CellProfiler. A systematic microscopy error, for example, could be detected at this point and further downstream data processing could be aborted without further investment of valuable time. In the absence of such egregious problems, it is helpful to store the QC results as metadata alongside subsequent analysis results to allow for retrospective quality checks and to assist troubleshooting. Alternately, the QC results can be used to exclude aberrant images from full analysis: if using CellProfiler for post-QC data analysis, for example, the classifier produced by the above steps may be incorporated into the CellProfiler analysis pipeline with the `FlagImage` module. This module can mark images that pass or fail QC and can also skip further analysis of images that fail and proceed immediately to the next image. Use the following steps to modify an existing CellProfiler analysis pipeline to take advantage of the QC results.
2. Load or create an analysis pipeline designed to score the assay of interest; examples of analysis pipelines may be found at http://www.cellprofiler.org/examples.shtml. If other modules are needed in the pipeline, they may be added and arranged using the controls at the bottom of the pipeline panel (*see* **Note 8**).
3. Select and add the `MeasureImageQuality` module from the "File processing" module category. Generally, this module should be placed as the first of the analysis modules.
4. Give this module the same settings as in the QC pipeline. Failure to do so may result in an error, as the same sets of features are expected between the two pipelines.
5. Select and add the `FlagImage` module from the "Data tool" module category. Place it in the pipeline after the `MeasureImageQuality` module. Adjust the settings listed in the following steps.
6. *Name the flag's category*: Leave this as "Metadata".
7. *Name the flag*: Give the flag a meaningful name. For example, if using this module to detect out-of-focus images, the flag might be called "OutOfFocus".
8. *Skip image set if flagged*: Select "Yes" for this setting to skip downstream modules in the pipeline for any images that are flagged. This approach gives the option of omitting unnecessary analysis on aberrant images. By selecting "No", the analysis measurements are retained regardless of the QC flag, which may be helpful for later review.

9. *Flag is based on*: Select the option from the drop-down box corresponding to the classifier file saved from CPA. Additional settings will appear prompting you to specify the location and file name of the classifier to be applied to this data set.
10. Further settings will allow you to select which classes to flag when applying the QC criteria, with the flag is set if the image falls into the selected class. The module can also set the flag if the image falls into any of multiple classes, e.g., if you created classes in CPA for both out-of-focus images and images with low cell counts as well as a class of in-focus images with high cellular confluency.
11. If you have multiple classifiers that indicate other QC problems (such as saturation artifacts in addition to focal blur), you can press the "Add another flag" button to produce another collection of settings for a new metadata flag; repeat the above steps to provide additional QC criteria.
12. Alternately, you can combine QC criteria to produce a single metadata flag by pressing the "Add another measurement" button to produce another collection of settings for the same metadata flag. Repeat the above steps to provide additional QC criteria and under the "*How should measurements be linked*?" setting, indicate whether the flag should be set if all the conditions are met ("Flag if all fail"), or any of the conditions are met ("Flag if any fail").
13. After running the pipeline on the full image set (follow the instructions for the QC pipeline in Subheading 3.8 above), the results of the classifier will be stored as a per-image measurement, named according to the settings for the `FlagImage` module. For example, with the example given in (Fig. 4), the corresponding measurement will be named "Metadata_OutOfFocus", with an out-of-focus image receiving a value of "1" while an in-focus image will assigned a value of "0".

4 Conclusions

This protocol describes how a researcher can collect a suite of image-based quality metrics and use a machine-learning approach to distinguish between high-quality and aberrant images, all with the use of free, open-source software. Naturally, the best approach to remove artifacts is to prevent them from occurring in the first place during sample preparation and imaging. Simple steps include filtering the staining reagents before use to remove large particulates, and confirming the proper exposure settings for each channel prior to running an experiment (*see* ref. 11). While we have taken a supervised (i.e., human-guided) approach, unsupervised (i.e.,

purely computer guided) techniques to whole-image classification have also been described (*see* ref. 12). We have restricted our guidance to out-of-focus images and images containing saturation artifacts because these classes cover nearly all the artifacts we typically see in our own experience, but this basic approach may be used to identify any desired artifact, provided that their "phenotype" can be captured by one or several whole-image measurements. If the measurements provided in the `MeasureImageQuality` module turn out to be insufficient to capture the artifact in question, it may be helpful to include additional measurements in the pipeline directed towards the artifactual features, analogous to what is done in the screening domain by including image features specific to the phenotype of interest (*see* ref. 13).

5 Notes

1. It is essential that the fluorescence images be collected with a uniform protocol, in which the image acquisition settings (e.g., exposure time, magnification, gain) are kept constant throughout the entire experiment. Additional guidance on image acquisition can be found elsewhere (*see* ref. 11). This workflow can be adapted to handle brightfield images as well, with the following caveat: debris will not appear as a saturation artifact but rather as a dark region or smudge. In this case, by inverting the pixel intensities so that dark pixels become bright, and vice versa, the quality control metrics described above can be used without modification. This can be done using the `ImageMath` module (Category: Image Processing) in CellProfiler with "Invert" as the *Operation* setting.
2. We recommend the use of "lossless" image formats such as .TIF, .BMP, or .PNG. While "lossy" .JPG images are commonly used for photography, the smaller file size comes at the cost of artifacts that can hinder image analysis. For further reading, please *see* the online Assay Guidance Manual chapter on image-based high content screening (*see* ref. 11).
3. If processing a few hundred images, a stand-alone desktop is sufficient to complete the task in a matter of hours. For assays with thousands of images or more, the best practice is to use a computing cluster to parallelize and thus speed up processing. Suggested hardware specifications for a computing cluster are 64-bit architecture, with eight or more cores per compute node.
4. Although most users of the protocol described in this article will not need source code, the source code is publicly available in Git repositories administered by GitHub, and can be downloaded from https://github.com/CellProfiler/CellProfiler/.

Information and resources for developers are available at https://github.com/CellProfiler/CellProfiler/wiki, including tips for running Cellprofiler on a cluster environment for large screens.

5. In addition to the Help menu in the main CellProfiler window, there are many "?" buttons in CellProfiler's interface containing more specific documentation. Clicking the "?" button near the pipeline window will show information about the selected module within the pipeline, whereas clicking the "?" button to the right of each of the module settings displays help for that particular setting. Additionally, the CellProfiler user manual is available online at http://www.cellprofiler.org/CPmanual/ (containing content copied verbatim from CellProfiler's help buttons), and a user forum (http://forum.cellprofiler.org/) is available for posting questions and receiving responses about how to use the software.
6. To add modules, click the "+" button below the pipeline panel (Fig. 3a). In the dialog box that appears, select the module category from the left-hand list. Select the module itself from the right-hand list. Double-click the module to add it to the pipeline, or click the "+ Add to Pipeline" button. Many modules can be added; click the "Done" button when finished. Modules can then be arranged in the pipeline by clicking the "^" or "v" buttons below the pipeline panel. Help is also available for each module by clicking the module to highlight it and then pressing the "?" button near the pipeline window.
7. By default, the `Images` module will pass all the files specified to later Input modules, in order to define the relationships between images and associated metadata (the `Metadata` module) and to have a meaningful name assigned to image types so other modules can access them (the `NamesAndTypes` module). Filtering the files beforehand is useful if, for example, a folder which was dragged-and-dropped onto the file list panel contains a mixture of images for analysis along with other files to ignore.
8. Often, the acquisition software of many screening microscopes will insert text into each image's file and/or folder name corresponding to the user-specified experiment name, plate, well, site, and wavelength number. For example, the BBBC images described above (*see* Subheading 2) use a common nomenclature, e.g., *Week1_150607_F10_s3_w1636CC6D1-0741-42BB-AF32-3785EB8BA086.tif*, where "Week1_150607" is the plate name, "F10" is the well, "s3" denotes site 3 in the well, and "w1" indicates that the first wavelength was acquired.

9. Regular expressions (regexp) are a versatile (albeit complex) text pattern-matching syntax. Patterns are matched using combinations of symbols and characters. By clicking the magnifying glass icon next to the regexp setting, a dialog is provided which shows a sample text string, a regexp, and the results of applying the regexp to the sample text; both the sample text and regexp can be edited by the researcher. CellProfiler's help text for the `Metadata` module provides an introduction to regular expressions. While regexp syntax is largely standardized, the Python programming language variant thereof is used here; a more in-depth tutorial can be found at http://docs.python.org/dev/howto/regex.html.
10. Raw grayscale images are recommended for fluorescence microscopy. If color images are acquired, select "Color image" for this setting, and, later, insert a ColorToGray module in the analysis portion of the pipeline. The `ColorToGray` module splits the original color image into its red, green and blue channels, each represented as grayscale image.
11. To use the metadata matching tool, select the metadata identifier that is required to uniquely match all the channels for each row. If multiple identifiers are needed, click the "+" button to add another row of metadata below the previous one, or the "−" button to remove a row. Click the up and down arrows to reorder the precedence that these identifiers are applied.
12. The QC metrics that are targeted to identify focal blur artifacts include: (a) Power spectrum slope: the image spatial frequency distribution, with lower values corresponding to increased blur; (b) Correlation: the image spatial intensity correlation computed at a given spatial scale offset, with lower values corresponding to decreased blur; (c) Focus score: the normalized image variance of the image, with lower values corresponding to increased blur; (d) Local focus score: the focus score computed in nonoverlapping blocks and averaged, with lower values corresponding to decreased blur. Details on robustness and prior validation of these metrics can be found elsewhere (*see* ref. 7).
13. The QC metrics that are targeted to identify saturation artifacts include: (a) Percent maximal: the percentage of the image occupied by saturated pixels; (b) Intensity standard deviation, which is useful for detecting images with very bright but subsaturated artifacts. Details on robustness and prior validation of these metrics can be found elsewhere (*see* ref. 7).
14. Measurements may reside in a MySQL or SQLite database. A MySQL database is recommended for storing large data sets (i.e., from more than 1000 images) or data that may need to be accessed from different computers. Consultation with the local

information technology staff on the details of setting up or accessing a database server is recommended. SQLite is another mode of data storage, in which tables are stored in a large, database-like file on the local computer rather than a database server. This is easier to set up than a full-featured MySQL database and is at least as fast, but it is not a good choice of storage if the data is to be accessed by multiple concurrent connections.

15. To prepare a pipeline for batch processing on a computing cluster, add the `CreateBatchFiles` module (Category: File Processing) as the last module of the pipeline and configure it according to the module instructions. Once done, click on the "Analyze images" button. CellProfiler will initialize the database tables and produce the necessary file for batch processing submission. Submit the batches to the computing cluster for processing; use the *Search Help...* function under the Help menu in CellProfiler to search for "batch processing" for details on cluster computing.
16. Click the PlateViewer icon in the CPA interface to launch a tool to view a single QC measurement as a per-well aggregate in a multi-well plate format. Use the ScatterPlot or Histogram tools for a more quantitative approach to reviewing single QC measurements. Details on the use of these tools for QC purposes can be found elsewhere (*see* ref. 7).
17. Each tile is a thumbnail of the full image; a small white square is displayed in the center of each tile as the mouse hovers over it. It may be that the image tile is too small to allow viewing a small or subtle artifact. One approach to handle this issue is opening the full image in a separate ImageViewer window by double-clicking the tile. From this window, the image can be placed into a bin by dragging and dropping the small white square in the image center. Another approach is to select "View" from the menu bar and in the dialog that appears, adjust the image zoom (indicated by the magnifying glass icon) by pulling the slider to the left, which will change the zoom of all the image tiles. Adjust until the image tiles are the desired size.
18. Images with no cells can usually be classified as in-focus for this purpose; enough residual cellular material often remains in such images for the microscope to maintain focus. Also, images with varying degrees of blurriness can all be included in the same bin for classification, as illustrated by the first two images in the "OutOfFocus" bin in Fig. 4.
19. Not all images in the "Unclassified" bin need to become part of the training set: if the classification of a particular image is uncertain, it can be ignored by leaving it in the "Unclassified" bin (or remove it by selecting it and pressing the Delete key).

Keep in mind, however, that Classifier will eventually be required to score all images in the experiment as one classification or the other, so the more information you provide in guiding this decision, the better.

20. The classifiers (except for Fast Gentle Boosting) are implemented using Python's scikit-learn package (*see* ref. 14). The values of the parameters used as input may be modified by selecting *Advanced* > *Edit Parameters*, but this is not recommended unless you are already comfortable with machine learning approaches.
21. The most accurate method to gauge Classifier's performance is to fetch a large number of images of a given class from the whole experiment, and evaluate the fraction of the images which correctly match the requested class. For example, if fetching 100 putative out-of-focus images reveals upon inspection that seven of the retrieved images are actually in-focus, then the classifier has a positive predictive value of roughly 93% (and thus a false positive value of 7%). Another approach is to click the "Evaluate" button to produce performance statistics (*see* **Note 22**); values closer to 1 indicate better performance. However, because the training set often includes a number of difficult-to-classify images (due to the recommended iterative training process), the accuracy reported by the "Evaluate" button should generally be considered the worst case scenario, that is, a lower bound on the true accuracy. The final approach is to open an image by double-clicking on an image tile and then select *Classify* > *Classify Image* to score the single image. While the results of this method cannot be extrapolated to other images, it can help improve a training set by identifying misclassified images to add to the classification bins; this can be done by left-clicking the full image and dragging-and-dropping it into the desired classification bin.
22. For both evaluation displays, the predictive ability is assessed using *cross-validation*, a technique in which the annotated set of images is split into a "training" subset to train the classifier to distinguish between classes, and a "test" subset to evaluate the accuracy. This procedure is repeated five times, with the training and test subsets randomly selected while preserving the percentage of samples for each class. The results are then aggregated to produce the evaluation displays. The confusion matrix shows a table in which the true classification of the images (rows) is shown versus the predicted classification (columns). Ideally, the table should have only non-zero values on the diagonal elements (i.e., where the row index = column index) and zeros elsewhere; this means that all images are correctly classified into their respective types. A large number of images in the off-diagonal elements indicates that the

classifier is "confusing" the classes with each other. The classification report displays a heatmap of three common metrics used in machine learning for each of the classes: the precision (how well the classifier avoided false positives, defined as the fraction of retrieved images for a given class that are correctly classified), the recall (how well the classifier obtained true positives upon request, defined as the fraction of correctly-classified images from the set of images retrieved) and the F1-score (a weighted average of the precision and recall; ranges from 0 to 1, with 1 as the best score). If using the classification report with the Fast Gentle Boosting classifier, the "Evaluate" button will produce a graph of the cross-validation accuracy for the training set, by estimating the classifier performance by training on a random subsample of the training set, then testing the accuracy on the samples not used for training. This value is plotted as an increasing number of image features are used. If the graph slopes upward at larger numbers of features, adding more features is likely to help improve the classifier. If the graph plateaus after a certain number of image features, then further features do not help improve accuracy. A downward slope may indicate more training examples are needed.

23. We have found that using only two classification bins to distinguish saturated from non-saturated images tends to fail for images that contain brightly fluorescing cells. This problem can be overcome by creating an additional class to distinguish bright, non-artifactual images from images containing actual saturated artifacts. If such images are unlikely to occur in a given assay, the creation and use of this extra bin can be omitted.
24. A saved classifier set can assist in initializing a QC classifier for a new experiment, as long as the stains and imaged channels are the same, as follows. Start a new CPA/Classifier session. Create and name the bins to match those from the previous Classifier session. Select *File* > *Load Classifier* from the Classifier menu to load your previously saved classifier. At this point, images can be fetched from the desired class without creating a training set first. The fetched images may have a large number of misclassifications, due to inter-experiment variability. If this is the case, the iterative workflow will still need to be followed as before.

Acknowledgements

This work was funded by the National Science Foundation (RIG DB-1119830 to M.A.B..) and the National Institutes of Health (R01 GM089652 to A.E.C.). We also thank Jane Hung and David Dao for offering helpful comments and suggestions during manuscript preparation.

References

1. Conrad C, Gerlich DW (2010) Automated microscopy for high-content RNAi screening. J Cell Biol 188:453–461
2. Thomas N (2010) High-content screening: a decade of evolution. J Biomol Screen 15:1–9
3. Niederlein A, Meyenhofer F, White D et al (2009) Image analysis in high-content screening. Comb Chem High Throughput Screen 12:899–907
4. Carpenter AE, Jones TR, Lamprecht MR et al (2006) CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol 7:R100
5. Kamentsky L, Jones TR, Fraser A et al (2011) Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. Bioinformatics 27:1179–1180
6. Jones TR, Carpenter AE, Lamprecht MR et al (2009) Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. Proc Natl Acad Sci U S A 106:1826–1831
7. Bray M-A, Fraser AN, Hasaka TP et al (2012) Workflow and metrics for image quality control in large-scale high-content screens. J Biomol Screen 17:266–274
8. Caie PD, Walls RE, Ingleston-Orme A et al (2010) High-content phenotypic profiling of drug response signatures across distinct cancer cells. Mol Cancer Ther 9:1913–1926
9. Ljosa V, Sokolnicki KL, Carpenter AE (2012) Annotated high-throughput microscopy image sets for validation. Nat Methods 9:637
10. Bray M-A, Carpenter A (2012) Advanced assay development guidelines for image-based high content screening and analysis. In: Sittampalam GS, Coussens NP, Nelson H et al (eds) Assay guidance manual. Eli Lilly & Company and the National Center for Advancing Translational Sciences, Bethesda, MD
11. Buchser W, Collins M, Garyantes T et al (2012) Assay development guidelines for image-based high content screening, high content analysis and high content imaging. In: Sittampalam GS, Coussens NP, Nelson H et al (eds) Assay guidance manual. Eli Lilly & Company and the National Center for Advancing Translational Sciences, Bethesda, MD
12. Rajaram S, Pavie B, Wu LF et al (2012) PhenoRipper: software for rapidly profiling microscopy images. Nat Methods 9:635–637
13. Logan DJ, Carpenter AE (2010) Screening cellular feature measurements for image-based assay development. J Biomol Screen 15:840–846
14. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830