

Screenit: Visual Analysis of Cellular Screens

Kasper Dinkla, Hendrik Strobel, Bryan Genest, Stephan Reiling, Mark Borowsky, Hanspeter Pfister

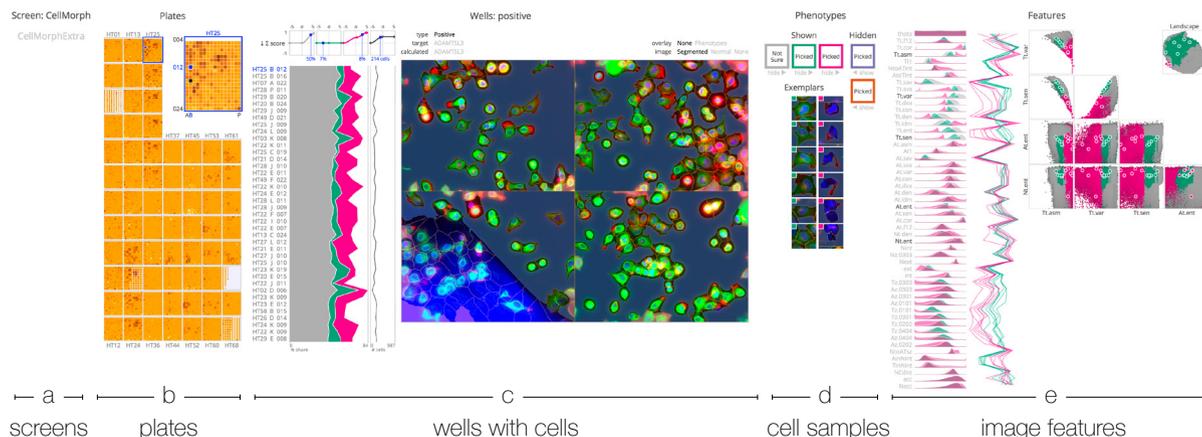


Fig. 1. An overview of Screenit with all columns opened: (a) The *screens* column shows the available screen data sets; (b) The *plates* column shows an overview of all plates in the selected CellMorph screen, where well score is shown with a color map that has dark colors for high scores and bright colors for low scores; (c) The *wells* column shows a hit list of top scoring wells at the left, and conditions and images of the selected well at the right; (d) The *phenotypes* column shows the cell phenotypes that are being modeled; (e) The *features* column depicts all cell image features as a list and provides information about feature value distributions.

Abstract— High-throughput and high-content screening enables large scale, cost-effective experiments in which cell cultures are exposed to a wide spectrum of drugs. The resulting multivariate data sets have a large but shallow hierarchical structure. The deepest level of this structure describes cells in terms of numeric features that are derived from image data. The subsequent level describes enveloping cell cultures in terms of imposed experiment conditions (exposure to drugs). We present Screenit, a visual analysis approach designed in close collaboration with screening experts. Screenit enables the navigation and analysis of multivariate data at multiple hierarchy levels and at multiple levels of detail. Screenit integrates the interactive modeling of cell physical states (phenotypes) and the effects of drugs on cell cultures (hits). In addition, quality control is enabled via the detection of anomalies that indicate low-quality data, while providing an interface that is designed to match workflows of screening experts. We demonstrate analyses for a real-world data set, CellMorph, with 6 million cells across 20,000 cell cultures.

Index Terms—High-content screening, visual analysis, feature selection, image classification, biology, multivariate, hierarchy

1 INTRODUCTION

The development of drugs is a process of trial and error. To discover compounds that could potentially cure disease, pharmaceutical companies test a wide spectrum of chemical compounds on cell cultures. Testing many different compounds at the same time is called a *screen*. Technological advancement enables larger and more complicated screens while reducing manual labor through automation. This shifts the bottleneck from the setup and execution of screens to the identification of promising compounds, which is referred to as *calling hits*. These hits are subject to *quality control* before being forwarded to biologists and chemists for elucidation and verification.

The physical setup of a screening experiment spans multiple levels, as shown in Fig. 2, where each:

Screen is divided across a number of plates that enable batch reads;
Plate embeds a grid of wells;
Well contains a solution of compounds and a group of cells, of which images are taken;
Cell has a physical state that is captured as numeric *image features*.

Therefore, the data derived from a screening experiment forms a multivariate hierarchy. These hierarchies, as well as the analysis goals of their users, pose an interesting visual analysis challenge. We present *Screenit* to support the visual analysis of these hierarchies. Screenit consists of tightly integrated visualizations that interweave the hierarchy levels and enable the interactive construction of models at multiple levels of the hierarchy. In this paper we carefully explore the goals and tasks for the analysis of screening experiments down to the cell level. We then present a design approach that can be generalized to suit multivariate hierarchies across application domains. Finally, we discuss the implementation of Screenit and demonstrate analyses for a real-world data set, CellMorph [10], with 6 million cells across 20,000 wells.

2 BIOLOGICAL BACKGROUND

Drug development is laborious and resource intensive. The discovery of a drug is usually not the direct result of a scientist who sees the potential of a specific compound to cure a specific disease for a specific, mechanistic reason. Instead, a large number of reasoned guesses are taken before the lead for a potential drug is discovered. In the past, a large number of technicians (or post-docs) were necessary to admin-

- Kasper Dinkla, Hendrik Strobel, and Hanspeter Pfister are with Harvard University. E-mail: kasper.dinkla@gmail.com, hstrobel@seas.harvard.edu.
- Bryan Genest, Stephan Reiling, and Mark Borowsky are with Novartis Institute of Biomedical Research. E-mail: bryan.genest@novartis.com, stephan.reiling@novartis.com, mark.borowsky@novartis.com.

Manuscript received 31 Mar. 2016; accepted 1 Aug. 2016. Date of publication 15 Aug. 2016; date of current version 23 Oct. 2016.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2016.2598587

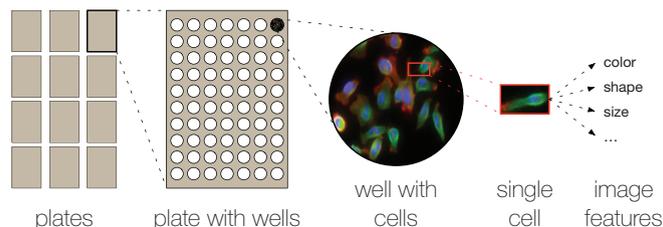


Fig. 2. The setup of a screening experiment consists of a collection of plates. Each plate contains a grid of wells, where each well contains a culture of cells that is exposed to a compound. Images are taken of each well and every cell is located and profiled to derive cell-specific image features.

ister a wide spectrum of chemical compounds to a large collection of cell cultures, and to subsequently observe the compounds' effects on cells. Recent technological advances have led to the automation of a large part of this screening process and the analysis of screening results [11, 39, 46]. Screens are now scaled to wider and more detailed compound spectra (high-throughput) while the observations of cell cultures are becoming more information rich (high-content).

2.1 High-throughput screens

High-throughput screening is the systematic, large scale screening of compounds within a short time span. Cell cultures are placed in a large number of wells (small Petri dishes) and a different compound is added to each well. These wells are embedded in plates to increase the number of wells that can be prepared and read simultaneously.

Coordinates Every plate contains a grid of wells with columns and rows. Each well is addressed in terms of plate, column, and row coordinates, where a plate is commonly assigned a short label with a number, a column is assigned a capital letter, and a row is assigned a number. For example, the well *HT46H010* is on plate *HT46* at column *H* and row *010*. Though wells contain cells, screeners do not refer to individual cells with coordinates, as described in Section 2.2.

Scale The size of screening experiments varies, but our collaborators regularly perform screenings that consist of hundreds of plates. These plates have 16 by 24 or 32 by 48 configurations, therefore containing 384 or 1536 wells, respectively. Each well contains up to hundreds of cells. This means that screens typically involve thousands of wells containing an aggregate of millions of cells.

Well conditions Wells are subject to more conditions than administered compounds. This includes the concentration of the compound, time point of administering the compound, time point of observing the well, and what equipment made the observations. Moreover, a single compound can be administered to multiple wells, either with a consistent compound concentration, referred to as *replicates* that enable screeners to check for consistent effects, or with increasing compound concentrations to extrapolate *dose-response curves* that capture the relationship between compound concentration and effect on cells.

Well features Conventional high-throughput screening methods do not capture information about individual cells but read aggregate signals per well, called well features. These signals are typically the intensity of light emitted by fluorescent compounds that have been added to the wells (in addition to the compounds that are screened for). These fluorescent compounds become active along with specific processes within cells. Capturing the emitted signals therefore enables the discrimination of cell physical states. The number of different fluorescent compounds that are used simultaneously is limited (below ten) by the ability to distinguish between the emitted light frequencies. Therefore, the output of high-throughput screening forms a multivariate hierarchy along the plate and well levels.

Observing aggregate signals at the well level is widely used because it is cost-effective. Its downside is limited resolution, where effects on individual cells within a well are not captured, causing two problems:

Effects on sub-populations of cells within a single well are missed, as their signal is lost in the average signal of the cell population;

Morphological effects are missed, including the physical shape and sub-cellular locations of fluorescent signals.

This information can be crucial for discriminating drug effects, which is why high-content screening is on the rise. We have designed Screenit to handle the scale of high-throughput experiments, with thousands of wells across dozens of plates, and to include high-content aspects with dozens of image features for millions of cells.

2.2 High-content screens

High-content screening relies on observation techniques that go beyond the observation of aggregate light from wells to detailed observations of individual cells within the wells [39].

Well images High-resolution microscopy images are taken of each well, where different fluorescent signals, called *channels*, are isolated per image. Sometimes multiple images are taken at different time points to track cell development [32].

Cell features summarize the physical state or shape of a cell as numbers. These numbers are derived automatically from well images via computational analysis called *profiling* [7]. First, individual cells are detected within the images to establish their coordinates and dimensions. Then many morphological features are computed, such as cell (or organelle) area, circumference, texture, and elongation. Our collaborators compute hundreds of these features for each cell and it is often unclear to them what the features actually quantify about a cell. A small number of these features end up being selected for the final analysis, which depends on the personal experience of the screener.

Cell phenotypes as well features A screener does not reason in terms of individual cells, but in classes of cells called phenotypes. A phenotype describes a general physical state of a cell. For example, a cell can be dying or about to divide into two. During screening the phenotype of a cell is inferred from its image features with classification methods from the machine learning domain [33]. For example, a classification algorithm could distinguish a dying cell from a dividing cell via an image feature that captures the cell's size. Here a small cell is likely to be dying, while a big cell is likely to be dividing and classified as such. Subsequently, the screener looks at the composition of phenotypes in a well to judge the effects of a compound. This mitigates some of the information loss incurred when looking only at the aggregate feature values of a well.

2.3 Bridging high-throughput and high-content

Now that high-content screening has become cost-effective, analysis of high-content information for high-throughput experiments has become a necessity. Screeners already use visualization tools to call hits and control quality, in part because calling hits depends on biological nuances that aggregate statistics do not always capture [2]. Likewise, quality control requires human detection of output errors that are hard to detect via automation. In fact, one of our screener collaborators stated that he never relies on statistical analysis alone to call hits and always verifies hits by looking for flaws in corresponding well (image) data and signal patterns across plates.

Analysis pipelines have expanded to cover the growing capabilities of measurement equipment. These pipelines feature visualizations that support hit calling and quality control, which includes high-content data. However, screeners tend to focus on a small number of familiar image features, thereby ignoring other features that could be of greater benefit [37]. Our collaborators recognize this problem and are building an analysis pipeline to manage the hundreds of image features that can now be profiled at the cell level. This pipeline filters the number of features down to dozens that have high entropy, based on negative (control) and positive wells. Nonetheless, a screener has to be involved in the subsequent analysis process to converge on a small number of relevant features—and the phenotypes that they discriminate. Existing visual analysis approaches support this effort but present the plate, well, and cell data aspects in isolation and without an overview.

This inhibits experts to easily navigate the data and quickly converge on relevant features, as described in the following section.

3 RELATED WORK

The individual analysis methods described in this paper are already in active use by the screening community. For example, cell profiling (inference of cell image features) is a large field with a number of established platforms [7,48]. The same holds for classifying cell images to infer phenotypes via (active) machine learning methods [19,33].

High-content analysis tools Several visual analysis tools already address the concerns described in Section 2.

Perkin Elmer's High Content Profiler is a proprietary visual analysis platform that supports complex screen setups at the plate and well level [2]. High-content features of cells are aggregated per well as feature distributions without fully exposing the cell level to the user. Hit modeling features dimension reduction, classification, and ranking at the well level. This is based on feature distributions of the wells, in which modeling and identifying cell phenotypes is not possible. Therefore, loss of detail occurs at the cell level, which impedes the derivation of hits at the well level.

Genedata's Screener is a proprietary platform like Perkin Elmer's but with strong integration of well images, in which multiple wells are easily selected and their images lined up for comparative analysis [1]. The cell level is exposed as scatter plots of cell populations for specific plates. Here the user can manually select an area in the plot to define a phenotype, and the abundance of this phenotype serves as a hit model for the plates' wells.

Phaedra is a new open source alternative [3]. It supports high-resolution well images and complex screening setups that can be shared across multiple experiments. Various plots can be generated for all levels, including cells. Phenotype modeling at the cell level is limited, as is hit modeling.

CellProfiler Analyst was the first platform that fully exposed the cell level in high-content analysis [20]. It features the definition of cell phenotypes via manual cell selection in plots (at all levels) and by training a classifier with user-labeled exemplar cells. This open source platform is designed for small screens setups and offers less functionality at the plate and well levels than the aforementioned platforms. For example, a navigable plate overview is missing and hit modeling is limited to ranking by a single feature or phenotype abundance.

HCS-analyzer is a more recent, open source platform for cell and phenotype analysis [27]. Well images have less importance in this approach, where the user does not pick and label cell exemplars based on the cell's image. Instead the cells of a user-selected well are clustered by their image features, where the user can label the clusters with phenotypes.

The first three platforms address screen analysis in the traditional high-throughput manner, in which large numbers of plates and wells have priority. The two latter platforms emphasize cell level and phenotype modeling. Screenit bridges these aspects by supporting large scale screening setups and cell level high-content analysis. All platforms share a design in which the screen levels have little navigation overlap. For example, plots are often requested for isolated features, cells, and wells, where moving up and down the plate, well, and cell hierarchy is a hurdle. Screenit integrates these levels in a consistent manner to enable fast navigation and cross-referencing between layers.

Multivariate models High-content screens contain data aspects that have been studied extensively in the visualization community, which has resulted in many advanced techniques for multivariate data visualization [21,40,47]. This includes the interactive exploration and manipulation of clustering algorithms and classifiers where experts gain knowledge and control of output models [6, 8, 17, 23, 26, 28, 41].

Multivariate hierarchies The visualization of hierarchies, or trees, is a thoroughly explored area [14,35]. The visual analysis of multivariate hierarchies also relates to the comparison of multiple hierarchies [12]. This includes the comparison of cell abundance in deep

taxonomies across different samples [9, 18]. Naturally, these techniques emphasize the topology of the hierarchy, communicating as many parent-child links as possible. Our design prioritizes conveying multivariate information at each level as encodings that our collaborating screeners prefer. Screenit therefore shows only the connection between the currently selected cell, well, and plate.

Compound effects Screenit is designed for data exploration in the early stages of hit analysis. However, as soon as a screener is confident about a compound's effect, this compound's hit information is transferred to other experts for more comprehensive analysis. Analyzing large quantities of compound substructure and cellular effect relations is challenging [44], where both machine learning [45] and visualization [22,29,38] approaches are beneficial. Beyond the structure of compounds, genetic information about compound targets (in case of RNAi for example) can be integrated to provide more context [16,24].

Complex screens High-content screening can also be used to track cell growth and duplication. Visualizing this data is a complicated problem, even when the hierarchy of the screening setup is ignored. Recently it has received attention in the visualization community [32]. Likewise, numerous parameters of image analysis algorithms affect the quality of screening data, where visualizations provide support as well [31].

4 ANALYSIS ROLES, GOALS, AND TASKS

During the first three months of the design process we interviewed each of four screening experts multiple times (including one author) to pinpoint their analysis goals and subsequently what visualization tasks Screenit has to support. We also took part in larger meetings with experts who offered analysis support for screening to nuance these goals and tasks. Each expert has a different expertise, plays a different role, and therefore has different priorities during analysis.

Biologists provide the motivation for performing screening experiments. They are knowledgeable about cell types, their disease-induced malfunctions and, therefore, their phenotype, the probable cellular mechanism by which to fix the malfunction, and compounds that could potentially affect this mechanism.

Chemists oversee the initial construction of a library of different compounds that could affect cells in a desired way. When an effective compound is discovered, chemists will be involved in pinpointing what aspects of a compound make it effective. This knowledge is then used to compose libraries that consist primarily of compounds that have these effective aspects. This process can be repeated several times to converge on compounds that have optimal effect, synthesis complexity, and side effects.

Bioinformaticians are statisticians or machine learning experts who are responsible for the numeric analysis of screens, which varies from screen to screen. This includes the development and tuning of cell profiling and phenotype classification algorithms.

Screeners are any of the above while also being responsible for setting up screens, calling hits, and quality control. After converging on high-quality hits, the screener will pass the hits—and context information—to biologists and chemists for further analysis.

These roles come with different analysis priorities but share the same goals:

GPM Phenotype Modeling is the creation and verification of a model that infers cell phenotypes from image features. This includes hypothesizing about what constitutes a phenotype.

GHM Hit Modeling is the creation and verification of a model that identifies compounds with a desired or interesting effect on the cell phenotypes in a well. This includes hypothesizing about what constitutes an effect.

GQC Quality Control is the elimination of unreliable data to reduce the number of wrongly called hits. This involves the identification of low-quality wells and plates.

From these domain goals we infer a set of data and visualization tasks that guide the design of our system (Section 5):

TNF Navigate and Filter Hierarchy to enable insights at different levels of detail, from the screen plates down to a single cell. Filtering applies to well conditions and cell phenotypes.

Goals: *GPM, GHM, GQC*

TFS Correlate, Inspect, and Select Image Features to choose those features that discriminate between potential phenotypes and minimize the complexity of the phenotype model. Goals: *GPM*

The tasks for modeling phenotypes (*GPM*) and hits (*GHM*) consist of iterations of *define model* \rightarrow *apply model* \rightarrow *visualize model* \rightarrow *re-define model*. Because this pattern is used differently for the phenotype and hit models, we describe different sets of tasks for both.

TPM Define a Phenotype Model by using the image features of example cells to train a classifier that discriminates phenotypes. Goals: *GPM*

TPA Apply Phenotype Model to infer a phenotype for every cell. Goals: *GPM*

TPV Visualize Phenotype Model to establish whether the model covers relevant phenotypes and classifies cells accurately. Goals: *GPM, GQC*

THM Define Hit Model via manipulation of a score function that weights phenotype abundance by desirability. Goals: *GPM*

THA Apply Hit Model to score and rank each well by the likelihood that it is indeed a hit. Goals: *GHM*

THV Visualize Hit Model to find those wells that are potential hits and to establish whether the model indicates relevant and accurate hits. Goals: *GHM, GQC*

5 ENCODING SCREEN LEVELS

We present a design that emerged through iterative prototyping and feedback sessions with screeners, spanning five months subsequent to establishing goals and tasks. Fully functional prototypes were deployed, used, and commented on by our collaborators during the final two months of the design process. This design features a column for each level of the screening experiment. These columns are laid out side by side, from left to right:

Screens shows a list of all available screens, where a single screen is selected for analysis, Fig. 1(a). This can be considered as an extra top level of the hierarchy that contains all screening experiments.

Plates provides an overview of all plates in the selected data set, Fig. 1(b). These plates have a static grid layout and all embedded

wells are colored according to their score in the hit model (*THV*). A plate (and well) can be selected in the overview, which is then depicted with more detail at the top right of the column.

Wells provides an overview of wells as a list at the left of the column, showing only wells that have a top score according to the hit model, which can be manipulated via a function editor at the top (*THM, THA, THV*). Detailed image and condition information of the selected well is shown at the right, Fig. 1(c).

Phenotypes shows the cell phenotypes that a screener wants Screenit to distinguish (*TPM*). Phenotypes are arranged into sub-columns, with an assigned phenotype color at the top and a list of exemplar cells at the bottom, Fig. 1(d).

Features relates phenotypes to image features for all cells (*TPA, TPV*), Fig. 1(e). This column is divided into three sub-columns: feature distributions as histograms, exemplar feature values as parallel coordinates, and feature distributions and exemplar values as a scatter plot matrix (*TFS*).

Most columns have the same arrangement, where an overview of elements is provided at the left side of the column, extending into a detailed view of select element(s) at the right side. This combination of simultaneous overview and detail provides a stable context for the screener to interact with. An alternative design would be a monolithic visualization of the hierarchy, such as a more abstract node-link depiction or tree map. However, early design iterations revealed that screeners reason in terms of the physical setup of an experiment, i.e., plates that embed wells and wells that embed cells. Therefore, we use the structure of the hierarchy levels to subdivide the interface into corresponding multiple linked views [34].

The combination of all columns span a great width where a typical computer setup with a single display is not able to show them all simultaneously. Luckily, it suffices for screeners to work either within a single column or across two adjacent columns at any time. Other columns are closed, until the user navigates to a column or performs an action that affects the contents of the column. This creates a configuration of sliding panes with two panes open in tandem at any time, an effective data exploration approach [42]. The header of a column is dimmed and tilted when its column is closed (shown in the accompanying video). Closed columns stack to provide the user a sense of position within the hierarchy. The prototype can also be adjusted to keep all columns opened simultaneously for larger display setups.

The plate, well, and cell (phenotype) levels are linked via a uniform selection model for consistent navigation (*TNF*). At most one plate, well, and cell are selected at the same time. Selections are always outlined blue, as seen for the selected plate and well in Fig. 1. This selection is constrained to the hierarchy, where the selected cell is always part of the selected well, and the selected well is always part of the selected plate. This navigation pattern matches the typical analysis workflows of screeners, described in the next section.

6 SCREENIT

We now walk through the analysis workflows of screeners, first navigating from left to right and then right to left across the columns. In addition, we discuss design details and decisions in light of early prototypes and feedback sessions.

6.1 Screen \Rightarrow Plates

The far left column shows a list of available screening data sets, shown in Fig. 1. The tool starts with a default selected data set, which in this case is the data set *CellMorph* that is described in Section 8. The selected data set is emphasized and shown within the header of the column, where it remains visible when the *screen* column is closed.

Plates The *plates* column opens as soon as a data set is selected. It provides an overview of all plates in the data set, in this case 68 plates named *HT01, HT02, ..., HT68*. These plates are stacked from top to bottom as columns, with first and last plate names shown at the top and bottom of the columns respectively.

Top and bottom plate names are displayed to orient the user but the names of the other plates are omitted to create a compact plate

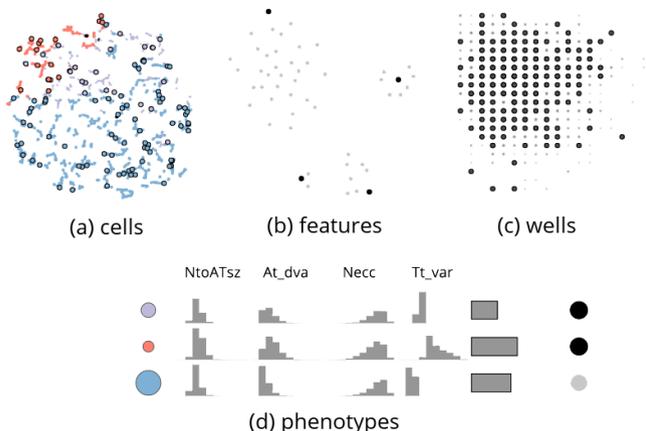


Fig. 3. An initial prototype that relies on abstract plots: (a) A sub-sample of cells is shown in a scatter plot, where every cell is encoded as a dot that is positioned according to its image features, and where its color represents inferred phenotype; (b) All features are encoded as dots that are positioned according to feature correlations; (c) All wells are encoded as dots that are positioned according to phenotype abundance; (d) Feature value distributions (columns) are encoded as histograms, partitioned by cell phenotypes (rows).

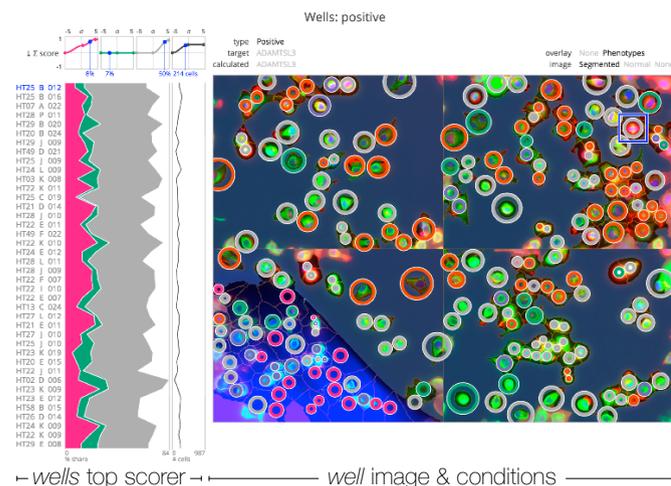


Fig. 4. The *wells* column in which well *HT25 B012* is selected (marked in blue) and its details are provided at the right side of the column: the well type is *positive*, the added compound targets *ADAMTSL3*, the displayed *image* is composed of four observations and is *Segmented*, and inferred cell *Phenotypes* are shown as an *overlay* of color coded circles. A list of top scoring wells is shown at the left, where phenotype shares are encoded as an area plot. An editor at the top of the hit list enables manipulation of the score model.

configuration. The display space that this requires is limited, where we assume the number of plates in a screening experiment to be below a hundred. This matches the smaller screening experiments of our collaborators. Every well can still be represented by a small number of pixels, avoiding visual aggregation of multiple wells.

This compact configuration eases comparison of well score patterns between plates. Likewise, the static configuration makes it easier for screeners to orient themselves and identify inter-plate well patterns that are related to the setup of the screen. All plates are shown in the overview with a static position to preserve the screener's mental map.

Heat maps Each plate in the overview shows the effect score of all of its wells with a color scale, where dark and bright colors indicate high and low scores, respectively. Initially, the score is defined to be high for wells that contain a large number of cells. Wells with a low number of cells therefore have a bright color. It is common practice to add a consistent number of cells to each well when setting up a screen. Big differences of cell counts between wells is therefore of interest.

For example, a well with a small number of cells suggests that many cells have died, which suggests that the added compound is toxic. In Fig. 1 we can already spot plates in column *HT25...HT36* with repetitive areas of dark, high-scoring wells. This suggests that the machine that consecutively read these plates could be defective (*GQC*). Alternatively, the added compounds could be laid out along the plates by increasing toxicity. These ambiguities can be resolved by navigating down the hierarchy to inspect individual wells in more detail (*TNF*).

In earlier prototypes we incorporated wells in more abstract visualizations, shown in Fig. 3(c). However, the screeners indicated multiple times that they prefer an overview of the wells that matches the physical setup of the plates. This overview includes individual well scores, where patterns at the plate level are less likely to be overlooked.

Detailed plate The *plates* column also shows a large representation of a single, selected plate at the top right. This detailed view eases the inspection of well scores and the selection of a specific well. It supports orientation by showing the plate, column, and row coordinates of the selected well and those at the boundary of the plate (Section 2.1). The outlining of the wells in the detailed plate bridges the more abstract well depiction of the plate overview.

6.2 Plates ⇒ Wells

The *wells* column is opened when a well is selected in either the plate overview or detailed plate view (*TNF*). An overview of top scoring wells, called the *hit list* is shown at the left of the *wells* column in Fig. 4. Right next to this list is an image of the selected well with associated information and options shown on top of the image.

Well image The image of the selected well takes up most of the *wells* column. When the *plates* and *wells* columns are opened this enables the navigation to (and validation of) those wells that seem to be suspect in the plate overview. At the top right of the image it is possible to switch between multiple images that have been captured of the same well. As seen in Fig. 4, two image types *Segmented* and *Normal* are available for the CellMorph screen:

Normal shows the integration of multiple measurements (fluorescent channels) into an image, in which activity in cell nuclei can be identified as blue, and the activity in the cytoplasm by green or red;

Segmented superimposes profiling information on the *Normal* image, demarcating cell and nuclei boundaries;

None does not show any image to make an overlay more legible.

In case of CellMorph, both image types consist of four sub-images captured at different spots of the same well. An overlay can be shown that depicts the cell phenotypes as colored circles. However, phenotypes first have to be modeled via the *phenotypes* and *features* columns.

Well conditions The conditions and type of a well are shown as a table above the well image. Wells can be annotated with conditions from multiple categories. For example, in Fig. 4 the *type* of the well is *positive* because gene *ADAMTSL3* is the *target* of an administered compound. This form of textual well annotation with little structure enables screeners to input a large variety of experiment setups.

Textual well conditions also tie in directly with the well filter functionality, where the user inputs part of a condition that a well should have. This filter is shown as part of the *wells* column header, for example *Wells: positive* in Fig. 4 where only wells with a *positive* condition are shown (*TNF*). Moreover, any of the conditions shown at the top left of the well image can be selected to make it the filtered condition. This enables fast lookup and cross referencing of well conditions, respectively. For example, in Fig. 1 the wells are filtered to match *positive*, where only those wells with a compound are shown, hiding the control wells. As a result the top row of each plate is lightened, as seen for row *004* of selected and enlarged plate *HT25*. The plates *HT03*, *HT22*, *HT66*, *HT68* show few positive wells.

An earlier prototype featured the ability to select multiple well conditions in the *wells* column, which then serve to partition the plates in the *plates* column according to what selected conditions are present on the plate. This change of plate layout confused the screeners and it did not fit in their current workflows. Instead, screeners preferred to filter for well conditions directly—at the plate level and the well level. This led to the current text filter, where the screener gets direct feedback when he is interested in comparing conditions.

6.3 Wells ⇒ Phenotypes

Any of the cells shown in a well image can be selected by clicking it, which opens the *phenotypes* column next to the *wells* column (*TNF*).

Phenotypes The phenotype model relates a cell's phenotype to its image features and it can be defined via the *phenotypes* column (*TPM*). The top of the column displays the phenotypes in the model, where there is a difference between those phenotypes that are currently shown and hidden, Fig. 1. Phenotypes are color coded via automated selection of colors from a fixed palette. In addition, phenotypes can be shown and hidden at any time by the user. Each shown phenotype has its own sub-column and is headed by a tile that shows the phenotype color and form:

Picked phenotypes are defined by exemplars that are handpicked by the user to shape a phenotype model;

Not Sure represents those cells that do not fit in the current phenotype model, i.e., a phenotype cannot be inferred with confidence.

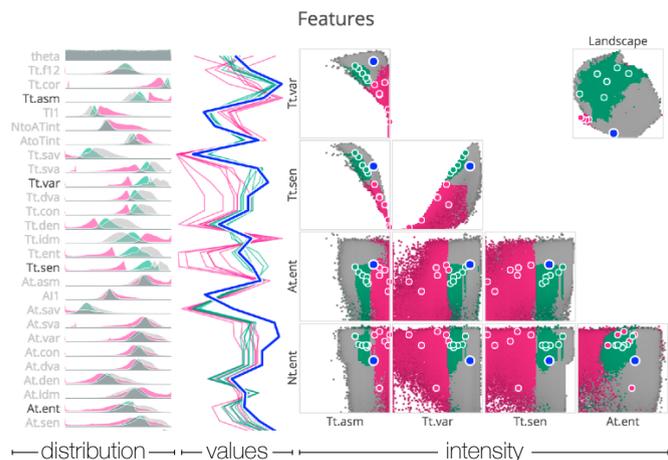


Fig. 5. Part of the *features* column, see Fig. 1 for full sized column. All image features are listed by name, dimming those that are not selected to be in the model. The three sub-columns next to it depict: feature value distributions as histograms, values per exemplar as parallel coordinates, and intensity plots of all combinations of selected feature pairs. In this case the intensity plots include a non-linear projection of all image features that forms a *landscape*.

The *picked* phenotypes have to be assembled by the user in the *exemplars* section. An enlarged picture of the selected cell is shown at the top, marked with a colored square at the top left to indicate what phenotype it has according to the current model. The user can confirm or counter this by adding the selected cell to the underlying exemplar stack of a phenotype. Alternatively, the selected cell can be assigned to a new phenotype. The exemplars' images form stacks, from which the user can distinguish the phenotypes, their assigned colors, and the visual criteria that motivated the user to define the phenotypes.

The exemplars and their user-assigned phenotypes form a training set from which a classifier (model) is derived. This classifier can evaluate the phenotype of any cell, from the cell's image features. Screenit uses a random forest classifier because it is robust and fast, but can easily be replaced by another classifier. Initially all of the screen's cells are evaluated to be of the *not sure* phenotype because there is no model without exemplar cells. Therefore, a typical workflow of one of our collaborator screeners is to first select a (negative) control well. In this well many cells are expected to be of a normal phenotype, because no compounds have been added that stimulate cells into an aberrant state. The screener then assembles these phenotypes by picking exemplars from the control well. These exemplars refine the model, where a proper phenotype will be evaluated for a larger number of cells.

By default the model is based on all available image features (CellMorph has 54 features). If possible, a screener wants to keep the phenotype model simple and therefore limits the number of selected features. However, selecting the right features is important to maintain the accuracy of the model. To make these informed decisions, more information about cells, corresponding inferred phenotypes, and their feature values is required.

6.4 Phenotypes \Rightarrow Features

Cell image feature information is acquired by opening the *features* column. The *features* column has three sub-columns, shown in Fig. 5.

Histograms The left column lists all cell features with their value distributions. These distributions cover the entire cell population partitioned by phenotype, shown as histograms that are colored according to their phenotype. These histograms enable the screener to spot features that could serve as phenotype discriminators. For example, a feature can have a bi-modal distribution or a distribution with a heavy tail. Both suggest groups of cells—potential phenotypes—where cells of each group have localized values for the feature in question.

The initial phenotype model uses all available features to distinguish phenotypes. However, when phenotypes and corresponding exemplars have been picked, as shown in Fig. 1, disparities between phenotype distributions indicate discriminating features—according to the phenotype model. For example, feature *Tl.var* in Fig. 5 has distinct distributions per phenotype, with strongly separated means. Discriminating features are potential candidates for inclusion in the model.

The *features* column has changed often during the design process, but the histogram list has been constant. Screenit exposes screeners to the richness of high-content screens, which is what the feature list provides at all times. It has been well-received during feedback sessions.

Selection A simplified model, based on a limited number of features, can be constructed by toggling the features in the feature list (*TFS*). Selected features have emphasized labels, as shown in Fig. 5. The phenotype model is redefined (*TPM*) and applied (*TPA*) as soon as the feature selection changes. The feature list is arranged according to pairwise feature correlations, where highly correlated features are positioned close to each other in the list. This enables a screener to select features with a high overall information content and it reduces clutter in the parallel coordinates plot.

The simplified model and its visualizations enables a screener to reason about modeled phenotypes in terms of features and the inducing biological mechanics within the cells. During a feedback session one bioinformatician screener emphasized the need to discover a screens' phenotypes but also to explain each phenotype, associate it with cellular mechanics and processes, and name it accordingly.

Parallel coordinates Adjacent to the feature distributions is a parallel coordinate plot that encodes the values of all features for every phenotype exemplar, shown in Fig. 5. The polylines are semi-transparent and colored by phenotype to make them easier to distinguish. The selected cell is drawn as a thick blue polyline, which enables the user to inspect individual exemplars by selecting them in the *phenotypes* column. As a complement to the feature histograms, the screener is therefore able to evaluate the exemplars that have been picked. For example, a screener can spot and select an outlying exemplar that has dubious feature values, or feature values that indicate an unexpected, novel phenotype (*TPV*).

Plot Matrix The selected features of a simplified model are shown as a matrix of intensity (2D histogram) plots on the right, as shown in Fig. 5. A plot is shown for every possible pair of selected features, where for every pixel the most abundant phenotype (for all cells in the data set) is identified by its assigned color. The brightness of the color relates to abundance. The low abundance areas are darker, where outlying cells are easier to spot, and the high abundance areas are brighter to indicate distribution peaks. This decomposes the phenotype model in terms of all cells and their feature values. This interests screeners who construct and tune phenotype classification algorithms.

The phenotype exemplars are shown as colored markers on top. These enable screeners to spot outlying exemplars that warrant inspection, such as the pink outlier in the bottom half of the matrix in Fig. 5. It also enables cross referencing phenotype exemplars and model distributions (*TPV*). For example, the model appears to make a clear cut between the green and pink phenotypes for the *Tl.asm* feature, in terms of the phenotype distributions of all cells, and the phenotype exemplars. But in the *At.ent* and *Tl.asm* plot we see that the distinction is not consistent for higher values of *At.ent*, where the classifier labels a large number of cells as *not sure*.

Initially we worked with scatter plots of a small sample (1000s) of cells, as shown in Fig. 3(a). The cells were given translucent colors according to their phenotype. However, these plots have poor legibility in areas where phenotype populations overlap. Alternatively, contour density plots were used for the intensity plots for some time. These plots are often used to visualize cell populations in Flow Cytometry experiments, which are familiar to screeners [15]. However, the contours created too much clutter when shown on top of each other for multiple phenotypes. The thin contours also inhibited screeners from distinguishing phenotypes by their colors.

Clutter is still an issue for the *features* column, as is the deceptive loss of information when reducing clutter in the plot matrix. However, the ability to hide and show phenotypes mostly mitigates these problems. This functionality was requested by one screener, who mentioned never needing to look at more than three phenotypes at a time.

Landscape Two special cell features can be included in a screen data set that are not part of the model but do form a landscape plot at the top right of the plot matrix. For the CellMorph data we pre-computed a multidimensional scaling of all cell image features via the t-SNE approach [43]. Cells with similar image features will therefore be positioned close to each other in the plot, in general. This provides insight into the high-dimensional distribution of phenotypes. For example, in Fig. 5 the exemplars of the pink phenotype have a dense configuration, which suggests that this phenotype covers a small and well-defined group of cells. This is not surprising since the exemplar images (Fig. 1) indicate that this phenotype models dead cells. On the other hand, the green phenotype appears to span a large number of cells across feature space. Again this comes as no surprise because the exemplar images indicate that the phenotype models healthy, dormant cells that make up a significant part of the numerous control wells.

6.5 Features ⇒ Phenotypes, Wells, Plates

Outlying cells and separate cell populations are easy to spot in these two dimensional histograms. For example, the *Tt.sen* and *Tt.asm* plot in Fig. 5 shows outlying cells to the left of the general cell population. The user is able to select any of the cells in a plot, immediately showing all of the cell's features in the parallel coordinate plot and its image in the *phenotypes* column. Subsequently, if the selected cell turns out to be of interest, it can be assigned to a phenotype as an exemplar. This constitutes an additional workflow to refine the phenotype model, where exemplars are picked according to their image features (and agnostic to well conditions) instead of their raw images (*TPM*).

Since the selected plate and well always conform to the selected cell, it is possible to navigate to the *wells* and *plates* columns to put the selected cell in context. For example, when an outlying cell is selected and the picture of the cell in the *phenotypes* column indicates that the cell has not been imaged properly, the screener navigates to its well and confirms that the entire well has a malformed image (*TNF*).

6.6 Phenotypes ⇒ Wells, Plates

A screener will navigate to the *wells* or *plates* column as soon as he is convinced that the phenotype model is sufficient to serve as a basis for future hit model(s). Usually this results in the *phenotypes* and *wells* columns opened. In the *phenotypes* column the screener can hide any phenotype that is not interesting. A hidden phenotype is excluded from the hit model, which effectively filters the hierarchy by cell phenotype (*TNF*). This also enables a screener to model those phenotypes that are currently of no interest (normal cells, for example). The screener will then proceed to the *wells* column to inspect the well hit list.

Hit List At the left of the *wells* column in Fig. 4 is a list of high scoring wells (*THV*). Wells are identified by their coordinates at the far left, where the selected well is always shown at the top in blue. Next to this are the abundances (or well shares) of the visible phenotypes, visualized as an area plot where phenotypes are easy to trace along the wells. Cell count is visualized at the right of the list as a thick black trend line. Both encodings concern related data and therefore match in style. However, the cell count encoding does not use filled areas to avoid confusion between the phenotype ratios and absolute cell counts. Fig. 4 has three visible phenotypes, pink, green, and gray for the *not sure* class. The selected well is always shown at the top of the hit list with a blue label, regardless of its hit score. Any of the wells in the hit list can be selected for detail on demand and navigation of the hierarchy (*TNF*). The *not sure* class is dominant across all top scoring wells, matching the hit function editor.

The hit list was not part of the design for a considerable time. We tried to visualize wells in terms of their phenotype abundance and without an explicitly defined hit score. For example, the high-dimensional projection of wells that is shown in Fig. 3(c) is similar to the static,

targeted analysis by bioinformaticians of the CellMorph data [10]. However, this did not appear to engage the screeners, except for one screener with a bioinformatics background. The screeners emphasized their use of *flower plots* to visualize phenotype abundance, and we therefore incorporated them, as shown in Fig. 6. Flower plots encode the phenotype shares of each well in a plate as arcs of fixed angle, where the arc area encodes phenotype abundance.

The flower plots required too much space to be legible when visualizing an entire plate and even though flower plots are often used by screeners, comparing phenotype abundances across an entire plate is not a common task. Filtering the flower plots by well conditions and user selection was included in the design to increase plot size and legibility, but instead it mostly disoriented the screeners. Moreover, screeners are used to composing hit lists that include context information (e.g., phenotypes) to pass on to chemists for more thorough analysis. Screeners asked for inclusion of such a hit list multiple times during the design process. Therefore, flower plots were succeeded by the area plot along the hit list, fulfilling both the need for a hit list and for information about phenotype abundance.

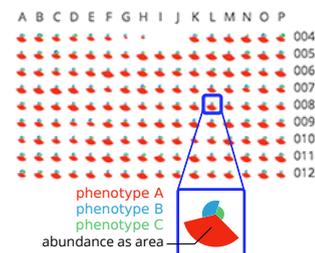


Fig. 6. In a *flower plot* phenotype shares are displayed per well as arcs of equal angles and varying area.

Hit Function Editor An editor is provided at the top of the well hit list that enables the screener to manipulate the hit model (*THV*). Each phenotype has its own plot with a curve that fits three control points that can be moved by the user. This curve is colored according to the phenotype that it represents and the ordering of the plots is consistent with the phenotypes in the hit list underneath.

Each curve defines the score contribution (y-axis) of a phenotype for a well, based on the ratio (x-axis) of the phenotype in that well. The domain of the curve is defined in terms of z-score, where the average phenotype share across all wells has z-score 0 at the origin of the plot (i.e., zero standard deviations from the mean). The minimum and maximum z-scores are limited to -5 and 5 respectively, beyond which the score is fixed. The score range spans $[-1, 1]$ along the y-axis. For a given well, the individual scores of all of its phenotypes, and its cell count, are summed to attain the aggregate well score.

For example, in Fig. 4 a hit score is defined where a well with an average to high cell count is promoted, a high share of *not sure* cells is promoted, and an average share of the pink phenotype is rewarded and a high share of the pink phenotype is highly awarded. Due to this hit model definition, we see that the top scoring wells in the list have consistently high shares of the pink and gray (*not sure*) phenotypes. In this case, screeners can find new phenotypes (*not sure*) that co-exist with the pink phenotype, such as selected well *HT25 B012* in Fig. 4. Editable transfer functions were requested repeatedly by one screener because simpler score models, such as a plain sum of abundances for a selection of phenotypes, were not expressive enough.

The score decomposition of the selected well is visualized explicitly, where every plot shows a blue dot that represents the selected well and its associated score. Absolute selected well share and cell counts are stated underneath the plots. Therefore, the screener is able to shape the hit function according to a particular well's phenotype shares. This enables the discovery of wells with (dis)similar phenotype shares.

The hit score editor was at one point part of the *phenotypes* column, where the editor plots shared the same sub-columns as the phenotypes and their exemplars. However, this results in a great distance between the editor (cause) and the heat map and hit list (effect). Placing the function editor on top of the hit list makes its function immediate, similar to the direct manipulation of column weights in a ranked list in LineUp [13]. It is therefore easier to see whether the edited function isolates a signal across the wells, as a skewed score distribution across the heat maps and as dark spots in Fig. 1.

Hit Score Overview Opening the plate view provides an overview of the well scores across the plates, as seen in Fig. 1. This reveals consistent hot spots for plates *HT21...HT30*, indicating similar cell effects due to consistent compound effects, or faulty observations. Moreover, this makes it easy for a screener to fine-tune the score function and reference the plate overview to determine whether he is converging on a clear signal (*THM, THV*).

7 IMPLEMENTATION

Screenit is web-based, where the client-side handles most visualization and interaction tasks, and the server-side handles screen data storage and analysis. The implementation is open source and publicly available at: <http://vcglab.org/screenit>

The client-side is written predominantly in TypeScript and renders visualizations via HTML5 canvas. Well information, such as abundance of phenotypes and cell counts, are stored client-side after being computed server-side. This enables scores to be computed client-side in a fraction of a second, which improves interactive editing of the score function and is feasible for up to 10,000s of wells. Likewise, well and cell images are fetched from the server when required. This keeps the memory usage of the browser below 1GB.

Cell computations are performed server-side, such as binning values per feature and phenotype, classifying cells, and subsequently deriving well phenotype abundance. The data served to the client is aggregated for wells and features, amounting to at most several MBs per request. The server-side data set is considerably larger. For example, the CellMorph data set has an image collection of over 100GB and the 54 image features of 6 million cells take over 1.5GB in compact form.

Image features have to be precomputed via a profiler [7] and stored as NumPy files, as described in the website. The server's analysis features are implemented in Python and rely heavily on packages Scikit-learn [30] and Pandas [25]. The default Scikit-learn random forest classifier is used to train and apply the phenotype model, but it can be replaced by an arbitrary classifier. The Tangelo framework handles server-client requests, which is agnostic of client state [4] where hardware can be scaled up and distributed to support more users.

Most analysis code is concurrent, such as cell classification and computing feature histograms. This enables the 6 million cells of CellMorph to be (re)analyzed in seconds and allows for future data sets to scale. Caching of computation results is supported to improve performance and response times. In case of CellMorph 16GB of server memory suffices to perform all computations while retaining all feature values in memory for optimal response times.

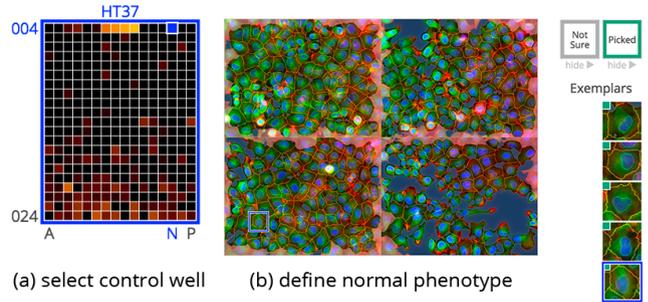
8 USE CASES

We present typical use cases of Screenit with an analysis of the CellMorph screen [10]. This screen consists of 6 million cells in 20,000 wells that are embedded in 68 plates. Each cell has 54 features, pre-computed and published by the CellMorph authors [10]. Each well contains a siRNA compound that disables a specific gene, which provides insight into the relation between gene activity and cell phenotypes.

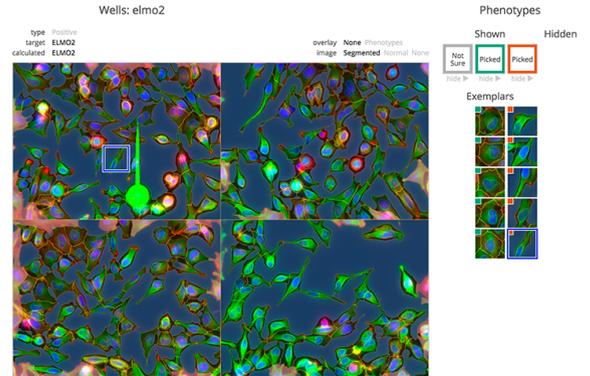
8.1 The case of ELMO's skeleton

The *ELMO2* gene is known to (indirectly) regulate the control of cytoskeleton organization. The cytoskeleton of a cell is a network of filaments that give a cell its rigid form and supports its internals. If the development, and dynamic alteration, of a cell's cytoskeleton is disrupted it impacts the physical shape of the cell. *ELMO2* is among the genes that are targeted and highlighted by the CellMorph study [10].

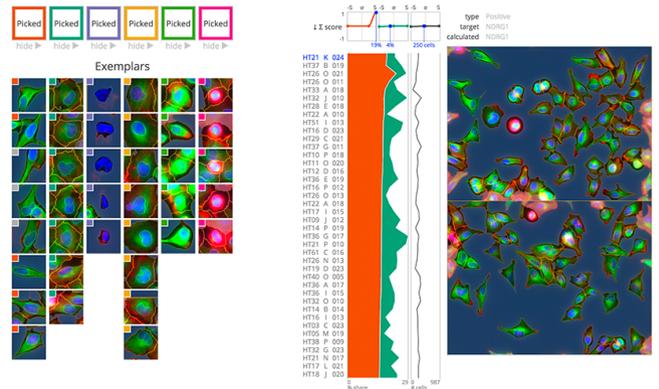
A point of reference To model the malformed cells that we expect for the *ELMO2* well, we first have to establish a basis of normal cells. We therefore select a well that we know to be a negative control, where no gene is affected. In the CellMorph screen the top row of every plate consists of negative control wells. We arbitrarily select the top well *HT37 N004* to navigate to a cell culture that consists of common phenotypes, shown in Fig. 7(a). This well is a rich source of normal cells from which we pick a number of exemplars that shape the first phenotype for the phenotype model, shown in Fig. 7(b).



(a) select control well (b) define normal phenotype



(c) define elongated phenotype



(d) refined phenotype model

(e) high scoring NDRG1 well

Fig. 7. *ELMO2* analysis steps.

ELMO's effect Next, we find the well where *ELMO2* is targeted by filtering for its well condition, shown in Fig. 7(c). The image of this well confirms the presence of many elongated cells, which can then be used to shape a second phenotype. The server subsequently trains the model, using the exemplars of our two phenotypes and all available cell image features. Navigating to the *features* column reveals the feature distributions across the entire population and shows which features discriminate the phenotypes that we are interested in. We select features *At.var*, *At.sen*, *Tt.var*, *Nt.den*, and *ext*.

Model refinement In the resulting density plots (not shown in the figures) both the normal and elongated cells span a broad range of feature values. Selecting outlying cells in the plots reveals novel phenotypes that are currently classified as normal or elongated. We therefore have to refine the phenotype model to narrow down what constitutes an elongated cell. Navigating up the hierarchy, to the wells that contain the novel phenotypes, reveals more cells that serve as extra examples. Using this approach, new phenotypes are added in iterations until all cells are divided amongst six phenotypes. This is a large share of the eight presented in the CellMorph study [10]. As seen in Fig 7(d)

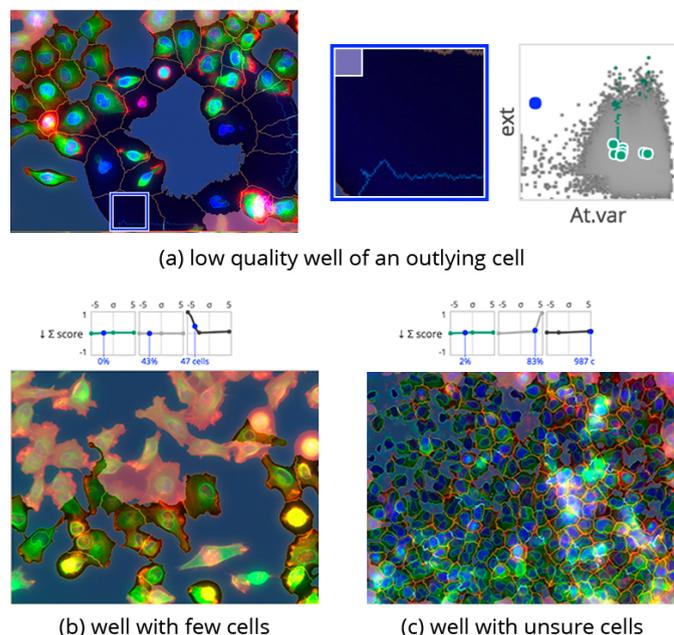


Fig. 8. Quality control steps.

for the refined model, the elongated cells (tagged orange) turn out to be especially hard to classify in comparison, as the model classifies three of the elongated exemplars to be *not sure* (tagged grey).

ELMO's affiliate The model's performance could be improved by adding more exemplars, but it turns out that we are already able to shape a hit model that identifies wells with a high share of elongated (or malformed) cells, shown in Fig. 7(e). Here a well is selected where *NDRG1* is targeted. Its well image reveals a large number of cells with a not necessarily elongated but odd shape. A lookup of *NDRG1* in UniProt reveals that the gene regulates microtubule dynamics, where microtubules are part of the cytoskeleton. This could explain the similarity of phenotypes between the *ELMO2* and *NDRG1* wells.

8.2 Quality control

Continuing from the ELMO use case, we walk through two quality control workflows. The first workflow starts at the cell level and the second at the plate overview. In the *phenotype* column we first hide all phenotypes except for the normal cells and the cells about which the model is *not sure*. Subsequently, outlying cells (with extreme feature values) are easy to spot. Selecting these cells reveals images that are of low quality. As shown in Fig. 8(a), selecting an outlying cell in the *ext* and *At.var* density plot reveals a cell that looks like a black blob. When we move up the hierarchy, to its well, we see in the well image that this cell is not an isolated case and that there could be a foreign object in the background that interferes with the imaging.

At the well level we are able to define a hit model, based on our refined phenotype model, which identifies potential low-quality wells. For example, an extremely low cell count can be rewarded. One well with a low cell count is shown in Fig. 8(b), where the image quality is so poor that the cell profiler has trouble detecting cells. Likewise, we look for wells with a high percentage of cells of the *not sure* phenotype in Fig. 8(c). This reveals a well with many cells that cannot be classified accurately, possibly caused by a bright patch in the image.

9 DISCUSSION

Screenit supports analysis of both high-throughput and high-content aspects of screens, within certain limits.

Bridging high-throughput and high-content Section 3 identifies a gap of priorities between existing screen visualization tools, in which industry platforms emphasize a well-centric analysis for large

screens, and academic tools emphasize cell-centric analysis for small screens. The design of Screenit bridges this gap by integrating plate, well, and cell levels side by side, in correspondence to screener workflows. This, for example, enables the context-preserving navigation of Section 8.2 from an outlying cell up to its containing wells and plates. Existing tools present the levels in greater isolation, where the position of a cell in the screen hierarchy is not immediately apparent.

In contrast with existing tools, Screenit provides detailed information for a large number of features at the cell level, almost treating features as another hierarchy level. Feature information is organized and displayed for all screen cells, where the feature values of a selected cell are apparent in context of all cells. Moreover, phenotype exemplars and inferred phenotypes of all cells are presented in this same context. This enables the fast identification and selection of discriminating features to construct a simplified model, and detailed model visualization to spot outlying cells for both features and phenotypes.

The image data that underlies high-content screens is strongly integrated into Screenit. Cells can be inspected as part of their well image. According to one screening expert this provides cell context and neighborhood information that is important for picking phenotype exemplars. Vice-versa, the well overlay of inferred phenotypes provides a context to assess the phenotype model. Existing tools either lack the cell images for exemplars selection, favoring feature value plots [27] with less context information, or show cell images in isolation [20].

Limitations While we designed Screenit to scale to a large number of cells, there are improvements to be made to increase the number of plates and cell features. Likewise, large screens have more than a hundred plates, which Screenit currently cannot support in an effective manner. Industry screening tools place more focus on plate and well configuration and analysis, featuring dose-response curve definition and visualization, and more elaborate well model visualization and manipulation. Likewise, Screenit focuses on the initial stage of screening QC and hit discovery, in which well meta-data is limited to plain tags. For now, compound structure and effect analysis, mentioned in Section 3, is performed afterwards with separate tools.

Lessons learned We were fortunate enough to collaborate with a large team of experts from different backgrounds, described in Section 4. This gave us a lot of input, albeit from experts with varying perspectives, which lead to mixed signals during large meetings. In particular, experts from various sub-domains had their own preferences for specific analysis and visualization methods. As a new experience, it required us longer than usual to pinpoint the analysis problem and subsequent goals. This happened as soon as we identified the screeners, or *front line analysts* [36], who we subsequently focused on.

Another new experience was the involvement of a dedicated business analyst (co-author) with an anthropology background and experience in requirements assessment for large projects. He set up small meetings with experts per sub-domain to ease determining individual goals. Moreover, he arranged meetings of different sizes and constellations, where we developed the design with a wide spread of feedback. All meetings were transcribed, which proved valuable for motivating subsequent design decisions and which is a known design aid [5].

10 CONCLUSIONS

We introduced Screenit, a visual analysis tool that enables screeners to inspect complex data from high-throughput and high-content experiments, and to create models for cell phenotypes and well hits. Screenit enables experts to spot data artifacts for quality control. By defining models and propagating model outcomes to the plate, well, and cell levels, screening experts can explore their data and modeling results in an integrated, holistic way.

ACKNOWLEDGMENTS

We thank our screening experts Eugen Lounkine, Ioannis Moutsatsos, and Fred Harbinski. We also thank James Tompkin for narrating the video. This work was supported in part by Novartis Institutes for BioMedical Research and the Air Force Research Laboratory and DARPA grant FA8750-12-C-0300.

REFERENCES

- [1] Genedata Screener. <http://www.genedata.com/products/screener>. Accessed: 06/20/2016.
- [2] Perkin Elmer High Content Profiler. <http://www.cambridgesoft.com/ensemble/spotfire/HighContentProfiler>. Accessed: 06/20/2016.
- [3] Phaedra. <http://www.phaedra.io>. Accessed: 06/20/2016.
- [4] Tangelo: a web application platform for Python programmers. <http://tangelohub.org>. Accessed: 03/20/2016.
- [5] M. Brehmer, J. Ng, K. Tate, and T. Munzner. Matches, mismatches, and methods: multiple-view workflows for energy portfolio analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 22(1):449–458, 2016.
- [6] S. Bremm, T. von Landesberger, J. Bernard, and T. Schreck. Assisted descriptor selection based on visual comparative data analysis. *Computer Graphics Forum*, 30(3):891–900, 2011.
- [7] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100, 2006.
- [8] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: an interactive visual analytics system for classification based on supervised dimension reduction. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 27–34. IEEE, 2010.
- [9] K. Dinkla, M. A. Westenberg, H. Timmerman, S. A. van Hijum, and J. J. van Wijk. Comparison of multiple weighted hierarchies: visual analytics for microbe community profiling. *Computer Graphics Forum*, 30(3):1141–1150, 2011.
- [10] F. Fuchs, G. Pau, D. Kranz, O. Sklyar, C. Budjan, S. Steinbrink, T. Horn, et al. Clustering phenotype populations by genome-wide RNAi and multiparametric imaging. *Molecular Systems Biology*, 6(1):370, 2010.
- [11] A. N. Goktug, S. C. Chai, and T. Chen. *Data analysis approaches in high throughput screening*, chapter 7. INTECH Open Access Publisher, 2013.
- [12] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.
- [13] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: visual analysis of multi-attribute rankings. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2277–2286, 2013.
- [14] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.
- [15] L. A. Herzenberg, J. Tung, W. A. Moore, L. A. Herzenberg, and D. R. Parks. Interpreting flow cytometry data: a guide for the perplexed. *Nature Immunology*, 7(7):681–685, 2006.
- [16] S. Heyse. Comprehensive analysis of high-throughput screening data. In *Proceedings of SPIE*, volume 4626, pages 535–547.
- [17] B. Hoferlin, R. Netzel, M. Hoferlin, D. Weiskopf, and G. Heidemann. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 23–32. IEEE, 2012.
- [18] D. H. Huson and N. Weber. Microbial community analysis using MEGAN. *Methods in Enzymology*, 531:465–485, 2013.
- [19] T. R. Jones, A. E. Carpenter, M. R. Lamprecht, J. Moffat, S. J. Silver, J. K. Grenier, A. B. Castoreno, et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proceedings of the National Academy of Sciences*, 106(6):1826–1831, 2009.
- [20] T. R. Jones, I. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter. CellProfiler Analyst: data exploration and analysis software for complex image-based screens. *BMC Bioinformatics*, 9:482, 2008.
- [21] J. Kehrler and H. Hauser. Visualization and visual analysis of multifaceted scientific data: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 19(3):495–513, 2013.
- [22] C. Kibbey, and A. Calvet. Molecular property explorer: a novel approach to visualizing SAR using tree-maps and heatmaps. *Journal of Chemical Information and Modeling*, 45(2):523–532, 2005.
- [23] J. Krause, A. Perer, and E. Bertini. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1614–1623, 2014.
- [24] P. Kumar, G. Goh, S. Wongphayak, D. Moreau, and F. Bard. Screensifter: analysis and visualization of RNAi screening data. *BMC Bioinformatics*, 14:290, 2013.
- [25] W. McKinney. pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 2011.
- [26] T. Muhlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the black box: strategies for increased user involvement in existing algorithm implementations. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1643–1652, 2014.
- [27] A. Ogier and T. Dorval. HCS-Analyzer: open source software for high-content screening data correction and analysis. *Bioinformatics*, 28(14):1945–1946, 2012.
- [28] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim. An approach to supporting incremental visual data classification. *Visualization and Computer Graphics, IEEE Transactions on*, 21(1):4–17, 2015.
- [29] C. Partl, A. Lex, M. Streit, H. Strobel, A.-M. Wassermann, H. Pfister, and D. Schmalstieg. ConTour: data-driven exploration of multi-relational datasets for drug discovery. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1883–1892, 2014.
- [30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. Scikit-learn: machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [31] A. J. Pretorius, M.-A. P. Bray, A. E. Carpenter, and R. A. Ruddle. Visualization of parameter space for image analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2402–2411, 2011.
- [32] A. J. Pretorius, I. A. Khan, and R. J. Errington. A survey of visualization for live cell imaging. *Computer Graphics Forum*, 2016.
- [33] P. Rämö, R. Sacher, B. Snijder, B. Begemann, and L. Pelkmans. CellClassifier: supervised learning of cellular phenotypes. *Bioinformatics*, 25(22):3028–3030, 2009.
- [34] J. C. Roberts. State of the art: coordinated & multiple views in exploratory visualization. In *Coordinated and Multiple Views in Exploratory Visualization, 2007*, pages 61–71. IEEE, 2007.
- [35] H.-J. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 17(4):393–411, 2011.
- [36] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2431–2440, 2012.
- [37] S. Singh, A. E. Carpenter, and A. Genovesio. Increasing the content of high-content screening: an overview. *Journal of Biomolecular Screening*, 19(5):640–650, 2014.
- [38] H. Strobel, E. Bertini, J. Braun, O. Deussen, U. Groth, T. U. Mayer, and D. Merhof. HITSEE KNIME: a visualization tool for hit selection and analysis in high-throughput screening experiments for the KNIME platform. *BMC Bioinformatics*, 13(Suppl 8):S4, 2012.
- [39] D. L. Taylor. *High Content Screening: A Powerful Approach to Systems Cell Biology and Drug Discovery*, chapter Past, Present, and Future of High Content Screening and the Field of Cellomics, pages 3–18.
- [40] C. Turkay, F. Jeanquartier, A. Holzinger, and H. Hauser. *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics: State-of-the-Art and Future Challenges*, chapter On Computationally-Enhanced Visual Analysis of Heterogeneous Data and Its Application in Biomedical Informatics, pages 117–140.
- [41] S. Van Den Elzen and J. J. Van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 151–160. IEEE, 2011.
- [42] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. *Computer Graphics Forum*, 32(3pt2):191–200, 2013.
- [43] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [44] M. Wawer, E. Lounkine, A. M. Wassermann, and J. Bajorath. Data structures and computational tools for the extraction of SAR information from large compound sets. *Drug Discovery Today*, 15(15–16):630–639, 2010.
- [45] M. J. Wawer, D. E. Jaramillo, V. Dančák, D. M. Fass, S. J. Haggarty, A. F. Shamji, B. K. Wagner, et al. Automated structure–activity relationship mining connecting chemical structure to biological profiles. *Journal of Biomolecular Screening*, 19(5):738–748, 2014.
- [46] R. Wollman and N. Stuurman. High throughput microscopy: from raw images to discoveries. *Journal of Cell Science*, 120(21):3715–3722, 2007.
- [47] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization: Overviews, Methodologies, and Techniques*, pages 3–33, 1997.
- [48] F. Zanella, J. B. Lorens, and W. Link. High content screening: seeing is believing. *Trends in Biotechnology*, 28(5):237 – 245, 2010.