The Image Data Explorer: interactive exploration of image-derived data

Coralie Muller[1], Beatriz Serrano-Solano[1,3], Yi Sun[1], Christian Tischer[2], Jean-Karim Hériché[1*]

[1]Cell Biology and Biophysics Unit and [2]Centre for Bioimage Analysis, European Molecular Biology Laboratory, Meyerhofstrasse 1, Heidelberg, Germany. [3]Current address: ELIXIR Germany, Institute of Bio- and Geosciences (IBG-5) - Computational Metagenomics, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

*Corresponding author

E-mail: heriche@embl.de (JKH)

Abstract

Many bioimage analysis projects produce quantitative descriptors of regions of interest in images. Associating these descriptors with visual characteristics of the objects they describe is a key step in understanding the data at hand. However, as many bioimage data and their analysis workflows are moving to the cloud, addressing interactive data exploration in remote environments has become a pressing issue. To address it, we developed the Image Data Explorer (IDE) as a web application that integrates interactive linked visualization of images and derived data points with exploratory data analysis methods, annotation, classification and feature selection functionalities. The IDE is written in R using the shiny framework. It can be easily deployed on a remote server or on a local computer. The IDE is available at https://git.embl.de/heriche/image-data-explorer and a cloud deployment is accessible at https://shiny-portal.embl.de/shinyapps/app/01_image-data-explorer.

Introduction

A typical bioimage informatics workflow for cellular phenotyping [1] involves segmentation (the identification of objects of interest in images, e.g. cells or nuclei) and feature extraction (the computation of measurements for each object) often followed by the application of a classifier to annotate the objects into biologically relevant classes. The use of a classifier requires the availability of already annotated objects to form the training set out of which the classifier algorithm will build a model. However, the generation of such a training set is often an iterative process that requires prior understanding of several aspects of the data. Such understanding is usually gained through the use of exploratory data analysis techniques such as plotting, dimensionality reduction and clustering. In addition, exploratory techniques are also useful for quality control (e.g. outliers identification) and curation of the data (e.g. identification of mis-annotated data points). Many image analysis software are available to perform segmentation and feature extraction with some also allowing annotation and classification [2]. Exploratory analysis of image-derived data is however generally left out of such software. Combinations of plugins for the popular image analysis software ImageJ [3] can be used for exploratory analysis. For example, the BAR plugin [4] contains routines for coloring regions of interests according to measurements. More extensive data exploration

tools were developed in the context of high content screening. Phaedra [5] and CellProfiler Analyst [6] are desktop-based applications that address exploratory needs by drawing interactive plots allowing visualization of images associated with data points. In addition, CellProfiler Analyst offers dimensionality reduction and classification functionalities while Phaedra can delegate machine learning functionalities to external scripts and workflow management systems.
Similarly, the Image Data Explorer (IDE) integrates linked visualization of images and data points with additional exploratory methods such as dimensionality reduction and clustering as well as classification and feature selection. Its light requirements on data input allows flexibility in data provenance. Another distinguishing feature is the possibility of deploying it as an online service as well as a desktop application.

Design and implementation

The Image Data Explorer is written in the R programming language [7] using the shiny framework [8]. This choice was motivated by two considerations. First, R gives access to a wide range of powerful statistical and machine learning methods with which the IDE functionalities could be extended should the need arise. Second, the shiny framework allows building web applications in R which, compared to desktop-bound applications, facilitates deployment and accessibility in a remote environment while still allowing easy installation for local use. The IDE is written using a modular architecture to facilitate addition of new functionalities.

The application requires two types of input: a set of images and a character-delimited text file of tabular data derived from the images. When the IDE runs on a user's computer (or a server), the images can be on any file system the user has access to from that computer, the only requirement is that the images must be under one common top-level directory (as opposed for example to being distributed across multiple file systems). Alternatively, the images can be put in an S3-compatible object store for remote access. The IDE can in principle read all image formats supported by the Bio-Formats library [9] but has only been extensively tested using multi-page TIFF files. The IDE can currently display images of at most 3 dimensions with the third dimension typically representing either depth (z coordinate) or time or channel. Images of higher dimensions are supported but in this case, the IDE merges the channels and the user has to decide which of depth or time should be displayed.

In the tabular data, rows represent either images or regions of interest (ROIs) in images and columns represent features and other associated data such as annotations of class membership. For the application to link data points to the corresponding images, the table should include at least one column with the path to image files relative to the image root directory or bucket of an S3-compatible object storage. ROIs are identified by the coordinates of an anchor point (e.g. the ROI centre) therefore there should also be a column for each of the relevant coordinates.

The user interface is a web page composed of different sections corresponding to the different functionalities accessible from a list on the left hand side of the screen. The entry point is the input section where the user can upload a tabular data file and configure the application to locate the images (e.g. by locating the image root directory on the server or giving object store connection details) and provide information on the table content (e.g. specify which column contains the relative path to the image files or which columns contain coordinates of ROIs anchor points). The input configuration parameters can be saved to a file which can be uploaded to the application to restore the configuration in a subsequent session.

The explore section (figure 1) is where the interactive visualization takes place. This section is divided into three areas: a plot viewer, an image viewer and a data table. These areas are linked in three way interactions such that selecting an element in any area highlights the corresponding data point in the others. The data table can be replaced by a second image viewer when two images need to be visualized for a given data point (for example, the original image and the corresponding segmentation mask). The plot viewer can be switched between a scatter plot, a histogram, a bar plot and a multi-well plate representation. The interactive plots are implemented using the ggplot2 [10] and plotly [11] libraries.
The image viewer is a modified version of the display function from the EBImage library [12].

The annotate section allows users to define labels and either create a new data table column or select an existing one as annotation-containing column. Selected data points can then be batch annotated with one of the available labels using a button in the explore section.

The dimensionality reduction and clustering sections allow application of various methods useful for finding patterns in the data. Methods currently implemented are principal component analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP,

[13] for dimensionality reduction (including supervised metric learning with UMAP) and k-means [14] and HDBSCAN [15] for clustering. The list is easily extensible to other methods. Dimensionality reduction methods add two new coordinate columns to the data table that are then used for plotting. Similarly, clustering methods add a column with cluster membership.

The classification and feature selection section implements gradient boosting classification via the XGBoost library [16]. It outputs various summary statistics on the model performance as well as a plot of feature importance which can be used to inform feature selection. A column with class predictions from the model can also be added to the data table. A button allows users to download the data table with all modifications made to it.

The statistics section gives access to one-way ANOVA and post hoc tests. The IDE assumes independence of the samples but otherwise tries to validate the assumptions underlying ANOVA. For sample sizes less than 30, the normality assumption is tested using the Shapiro-Wilk test and if the data is deemed not normally distributed, ANOVA is performed using the non-parametric Kruskal-Wallis test and post hoc tests using pairwise Wilcoxon rank sum tests. The homogeneity of variance is tested using Bartlett's test and when unequal variances are detected, the IDE performs Welch's ANOVA. Post hoc tests are then carried out with the Games-Howell test.

Results

Mitotic chromosome condensation and nucleolus formation are two processes thought to be driven by phase separation [17, 18]. Proteins involved in these two processes could be candidate regulators of phase separation. To explore this idea, we sought to analyse nucleoli from images of gene knockdowns known to affect chromosome condensation. To this end, we processed publicly available images from a published RNAi screen for chromosome condensation [19] with CellProfiler using a Galaxy [20] workflow to characterize nucleoli (workflow available on the WorkflowHub at https://workflowhub.eu/workflows/41). In brief, chromosome condensation hits known to localize to the nucleolus were selected as well as NPM1, a nucleolar protein whose knockdown is known to disrupt nucleolar structure [21], and corresponding images were accessed from the Image Data Resource [22] (https://idr.openmicroscopy.org/webclient/?show=screen-102), nuclei were segmented out of the DNA channel at time point 170 and nucleoli segmented as “holes” in each nucleus. Features were extracted for both nuclei and nucleoli and nucleoli features averaged for each nucleus. This resulted in a data table in which each row represents a segmented nucleus with summary descriptors of its nucleoli. We use the IDE to explore whether any gene knockdown resulted in a nucleolar phenotype.

We start by generating a scatterplot of nucleus size vs nucleoli number. This reveals that larger nuclei tend to have more nucleoli. Selecting outlying points allows for visual inspection of the corresponding nuclei in the original image as well as in the segmentation mask (figure 2) and reveals that although nuclei have generally been adequately segmented, very large nuclei correspond to segmentation artefacts. Selecting the siRNA column as group variable and switching to a box plot representation of nucleoli count (figure 3A) shows that control cells typically have 1-3 nucleoli with a median of 2 nucleoli while some knockdowns seem to increase the proportion of cells with 0 or 1 nucleolus. Among these are the two siRNAs targeting NPM1. Nuclei with a high number of nucleoli are consistent with the larger number of nucleoli observed after mitosis [23]. We next asked if other features could prove useful to detect nuclei with nucleolar phenotypes. To explore this, we switch to the annotate section and create an annotation column with two labels: control and phenotype, then back in the explore section, we select NPM1 nuclei from the siRNA with the strongest effect on nucleoli number and annotate them as phenotype. To find negative controls from neighbouring wells, we make use of the table search functions to filter the rows by plate, well and/or siRNA ID and annotate negative control nuclei as control. We then switch to the classification and feature selection section to train a XGboost classifier on all features to produce a plot of relative feature importance (figure 3B) where the values over all features sum to 1. The feature importance reported by the IDE is the gain which is a measure of the improvement in accuracy contributed by a feature [16]. When comparing two features, the one with the highest value is more important for the classifier performance. However, before checking feature importance, we ensure that the classes can be distinguished. For this, the IDE reports several statistics on the classifier performance. In this case, the resulting classifier has an accuracy of 74% which can be compared to the no information rate which is the fraction of the most abundant class and corresponds to the best performance that can be obtained by a naive classifier always assigning the most common label to samples. Confidence in the accuracy being above the no information rate is given by the low associated p-value of 8e^{-9}. Here, the most significant feature corresponds to nucleolus size. A box plot of nucleolus size distribution by siRNA confirms this as negative controls and some siRNAs show a median nucleolus size around 40 pixels with some siRNAs, including those targeting NPM1, having a median around 20.

Our quick exploration with the IDE thus indicates that images from this data set contain relevant information and already identifies a few knockdowns that may warrant further analysis. However, it also suggests that derived features such as the fraction of nuclei with certain properties (e.g. nucleolar size above or below some threshold) could be useful to more robustly identify nucleolar phenotypes.

Discussion

The IDE couples exploratory data analysis methods with interactive visualization of images and associated data. The light input requirements make it ideally suited for quick exploration of image data from different sources because many image analysis tools can produce character-delimited tabular data. In its current implementation, it is particularly suitable for small to medium-sized data sets (up to hundreds of thousands of data points). The modular architecture makes it easy to add new tools and its implementation as a web application makes it easily deployable in cloud environments for remote access.

Availability and future directions

The IDE is licensed under the GNU General Public License v3.0. The code is managed with a GitLab instance at https://git.embl.de/heriche/image-data-explorer from which it can be downloaded and where documentation can be found in the wiki section. A cloud-based deployment is available here: https://shiny-portal.embl.de/shinyapps/app/01_image-data-explorer. Future work will include support for images in OME-NGFF, the next-generation file format for bioimaging [24] to improve interoperability and facilitate more flexible data access, for example by allowing direct access to images from repositories such as the IDR or the BioImage Archive [25]. With a foreseen increase in the use of the Galaxy platform for image analysis, we plan to deploy the IDE as a Galaxy interactive environment to give it direct access to data produced by Galaxy workflows.

Supporting information

S1 contains the IDs of images retrieved from the IDR.

S2 contains the table with data on nuclei and nucleoli used as input to the IDE to produce figures 2 and 3.

S3 contains an example of saved input parameters including connection to the image object storage. Alternatively, the associated images can be accessed by the IDE by manually selecting as image root directory the S3 bucket named screens at end point s3.embl.de.

Author contributions

CT and JKH conceived the project, CM and JKH wrote the code, YS deployed it in the cloud, BSS contributed data and analysis

Acknowledgements

We thank Aliaksandr Halavatyi, Kimberly Meechan and Hugo Botelho for discussions and suggestions and Valerie Petegnieff for test data.

References

1. Sommer C, Gerlich DW. Machine learning in cell biology - teaching computers to recognize phenotypes. J Cell Sci. 2013; 126(Pt 24):5529-39. doi: 10.1242/jcs.123604.

2. Smith K, Piccinini F, Balassa T, Koos K, Danka T, Azizpour H, et al. Phenotypic Image Analysis Software Tools for Exploring and Understanding Big Image Data from Cell-Based Assays. Cell Syst. 2018; 6(6):636-653. doi: 10.1016/j.cels.2018.06.001.

3. Rueden, CT, Schindelin J, Hiner MC, DeZonia BE, Walter AE, Arena ET, et al. ImageJ2: ImageJ for the next generation of scientific image data. BMC Bioinformatics 18, 529 (2017). doi: 10.1186/s12859-017-1934-z

4. Ferreira T, Miura K, Bitdeli Chef, Eglinger J. Scripts: Bar 1.1.6. (2015, August 21). Zenodo. doi:10.5281/ZENODO.28838.

5. Cornelissen F, Cik M, Gustin E. Phaedra, a protocol-driven system for analysis and validation of high-content imaging and flow cytometry. J Biomol Screen. 2012; 17(4):496-506. doi: 10.1177/1087057111432885.

6. Stirling DR, Carpenter AE, Cimini BA. CellProfiler Analyst 3.0: Accessible data exploration and machine learning for image analysis. Bioinformatics. 2021 Sep 3:btab634. doi: 10.1093/bioinformatics/btab634.

7. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Available from https://www.R-project.org/.

8. Chang W, Cheng J, Allaire JJ, Sievert C, Schloerke B, Xie Y, et al. shiny: Web Application Framework for R. R package version 1.7.1. Available from https://CRAN.R-project.org/package=shiny

9. Linkert M, Rueden CT, Allan C, Burel JM, Moore W, Patterson A, et al. Metadata matters: access to image data in the real world. J Cell Biol. 2010; 189(5):777-82. doi: 10.1083/jcb.201004104.

10. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

11. Sievert C. Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC Florida, 2020.

12. Pau G, Fuchs F, Sklyar O, Boutros M, Huber W. EBImage--an R package for image processing with applications to cellular phenotypes. Bioinformatics. 2010;26(7):979-81. doi: 10.1093/bioinformatics/btq046.

13. McInnes, L, Healy, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv:1802.03426 [Preprint]. 2018. Available from https://arxiv.org/abs/1802.03426.

14. Hartigan JA, Wong MA. Algorithm AS 136: A K-Means Clustering Algorithm. Applied Statistics. 1979; 28(1):100-108. doi: 10.2307/2346830

15. Hahsler M, Piekenbrock M, Doran D. dbscan: Fast Density-Based Clustering with R. Journal of Statistical Software. 2019; 91(1): 1-30.

16. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. p. 785–794. doi:10.1145/2939672.2939785

17. Strom AR, Brangwynne CP. The liquid nucleome - phase transitions in the nucleus at a glance. J Cell Sci. 2019; 132(22).

18. Cuylen S, Blaukopf C, Politi AZ, Müller-Reichert T, Neumann B, Poser I, et al. Ki-67 acts as a biological surfactant to disperse mitotic chromosomes. Nature. 2016; 535(7611):308-12.

19. Hériché JK, Lees JG, Morilla I, Walter T, Petrova B, Roberti MJ, et al. Integration of biological data by kernels on graph nodes allows prediction of new genes involved in mitotic chromosome condensation. Mol Biol Cell. 2014; 25(16):2522-2536.

20. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Cech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018; 46(W1):W537-W544. doi: 10.1093/nar/gky379.

21. Holmberg Olausson K, Nistér M, Lindström MS. Loss of nucleolar histone chaperone NPM1 triggers rearrangement of heterochromatin and synergizes with a deficiency in DNA methyltransferase DNMT3A to drive ribosomal DNA transcription. J Biol Chem. 2014; 289(50):34601-19. doi: 10.1074/jbc.M114.569244.

22. Williams E, Moore J, Li SW, Rustici G, Tarkowska A, Chessel A, et al. The Image Data Resource: A Bioimage Data Integration and Publication Platform. Nat Methods. 2017; 14(8):775-781.

23. Savino TM, Gébrane-Younès J, De Mey J, Sibarita JB, Hernandez-Verdun D. Nucleolar assembly of the rRNA processing machinery in living cells. J Cell Biol. 2001; 153(5):1097–1110.

24. Moore J, Allan C, Besson S, Burel JM, Diel E, Gault D, et al. OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. Nat Methods. 2021; 18(12):1496-1498. doi: 10.1038/s41592-021-01326-w.

25. Ellenberg J, Swedlow JR, Barlow M, Cook CE, Sarkans U, Patwardhan A, et al. A call for public archives for biological image data. *Nat Methods.* 2018; 15(11) 849-854. doi:10.1038/s41592-018-0195-8.

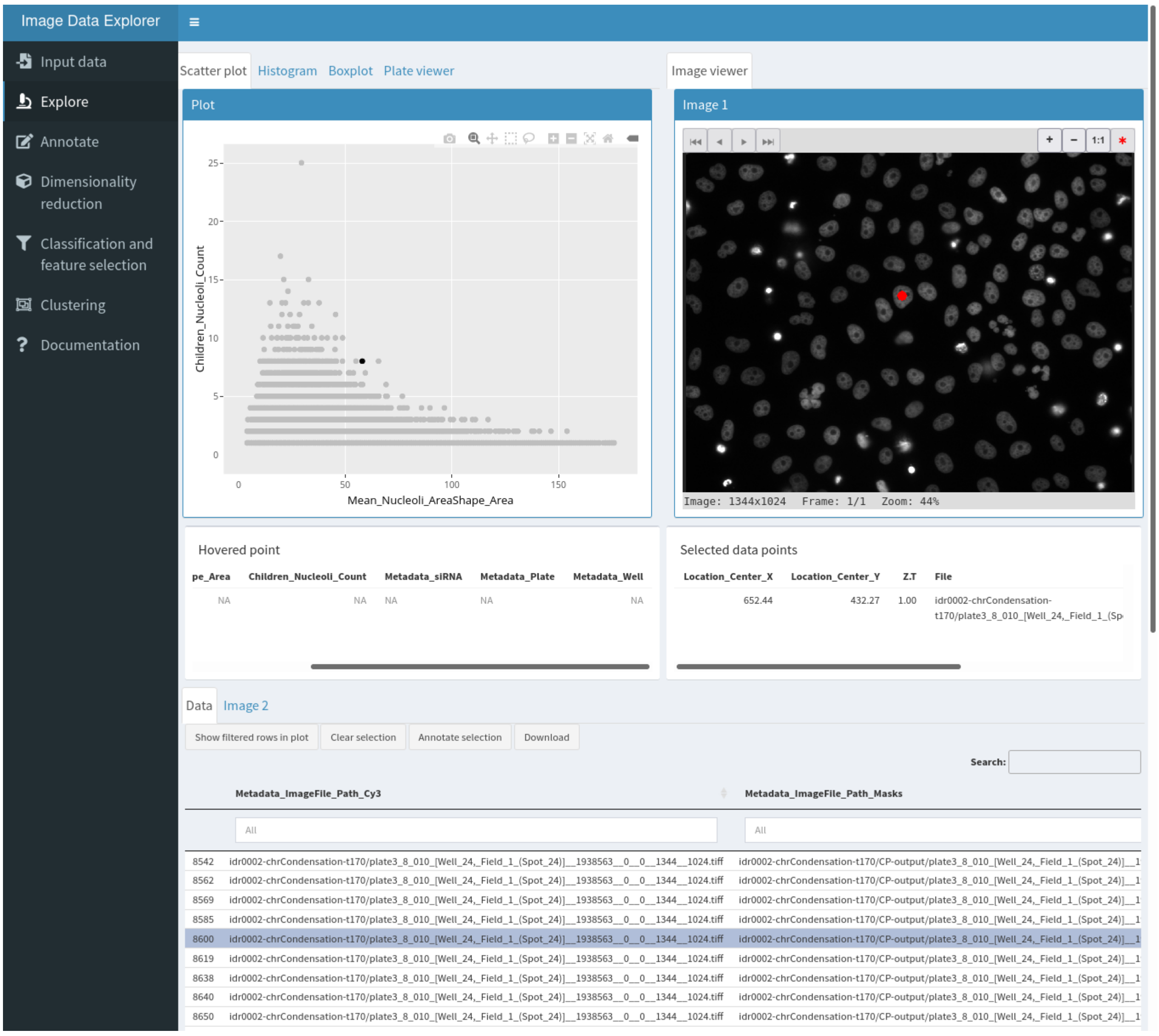
Figure legends

Figure 1: The explore workspace. The top left panel is used for plotting, the top right panel is an interactive image viewer and the bottom section shows the interactive data table with a tab to access a second image viewer. The red dot in the image viewer indicate the position of the segmented object corresponding to the selected data point shown in black in the scatter plot and highlighted in the data table.

Figure 2: Use of a second image viewer to simultaneously visualize the original image and the segmentation mask corresponding to a selected data point.

Figure 3: A - Box plot of nucleolar number per siRNA as produced by the IDE. Each colour corresponds to data from one siRNA. B - Output of the XGBoost classifier. Left panel: Plot of relative feature importance, colours are used to indicate clusters of features with similar importance. Right panel: Information to assess performance of the trained classifier.

Figure 1



```
Image: 1344x1024   Frame: 1/1   Zoom: 44%
```

Hovered point

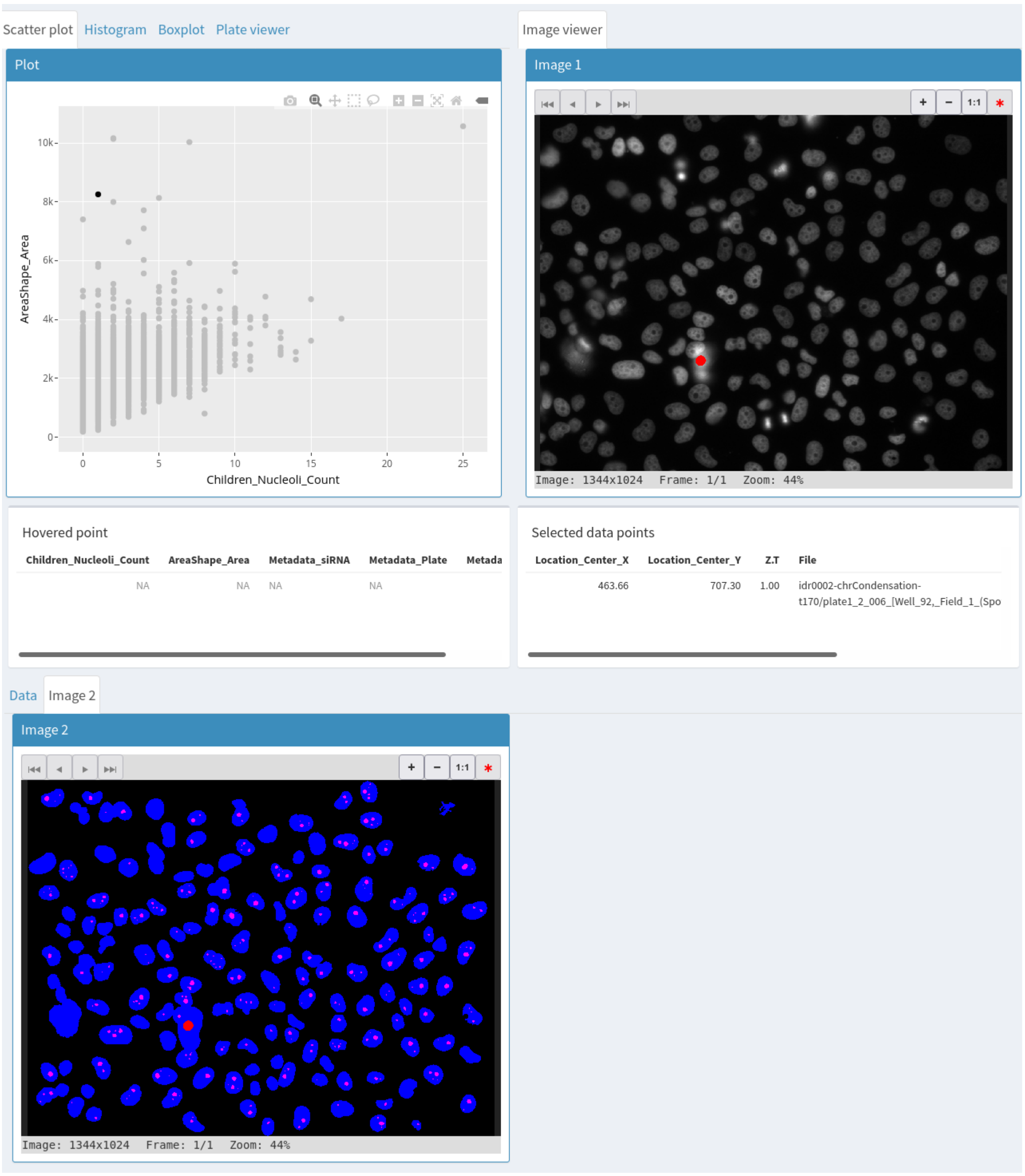
pe_Area	Children_Nucleoli_Count	Metadata_siRNA	Metadata_Plate	Metadata_Well
NA	NA	NA	NA	NA

Selected data points

Location_Center_X	Location_Center_Y	Z.T	File
652.44	432.27	1.00	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Sp

	Metadata_ImageFile_Path_Cy3	Metadata_ImageFile_Path_Masks
	All	All
8542	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8562	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8569	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8585	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8600	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8619	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8638	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8640	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1
8650	idr0002-chrCondensation-t170/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1938563__0__0__1344__1024.tiff	idr0002-chrCondensation-t170/CP-output/plate3_8_010_[Well_24,_Field_1_(Spot_24)]__1

Figure 2



Hovered point

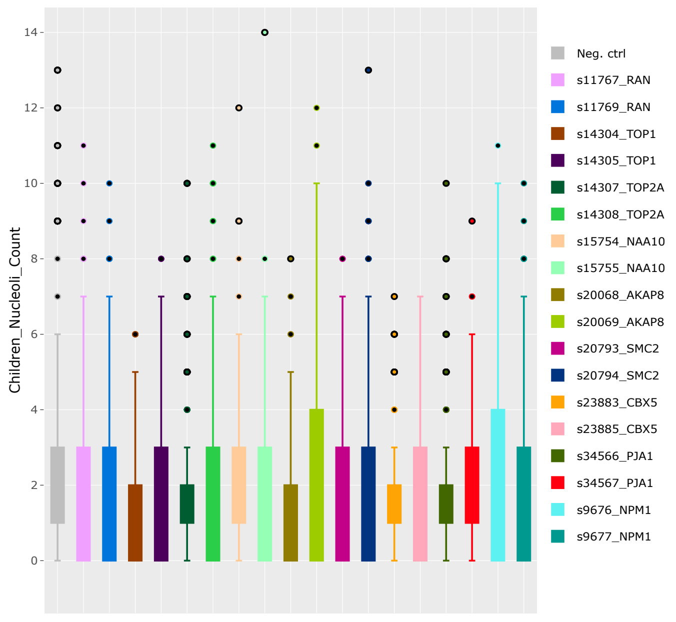
Children_Nucleoli_Count	AreaShape_Area	Metadata_siRNA	Metadata_Plate	Metada
NA	NA	NA	NA	

Selected data points

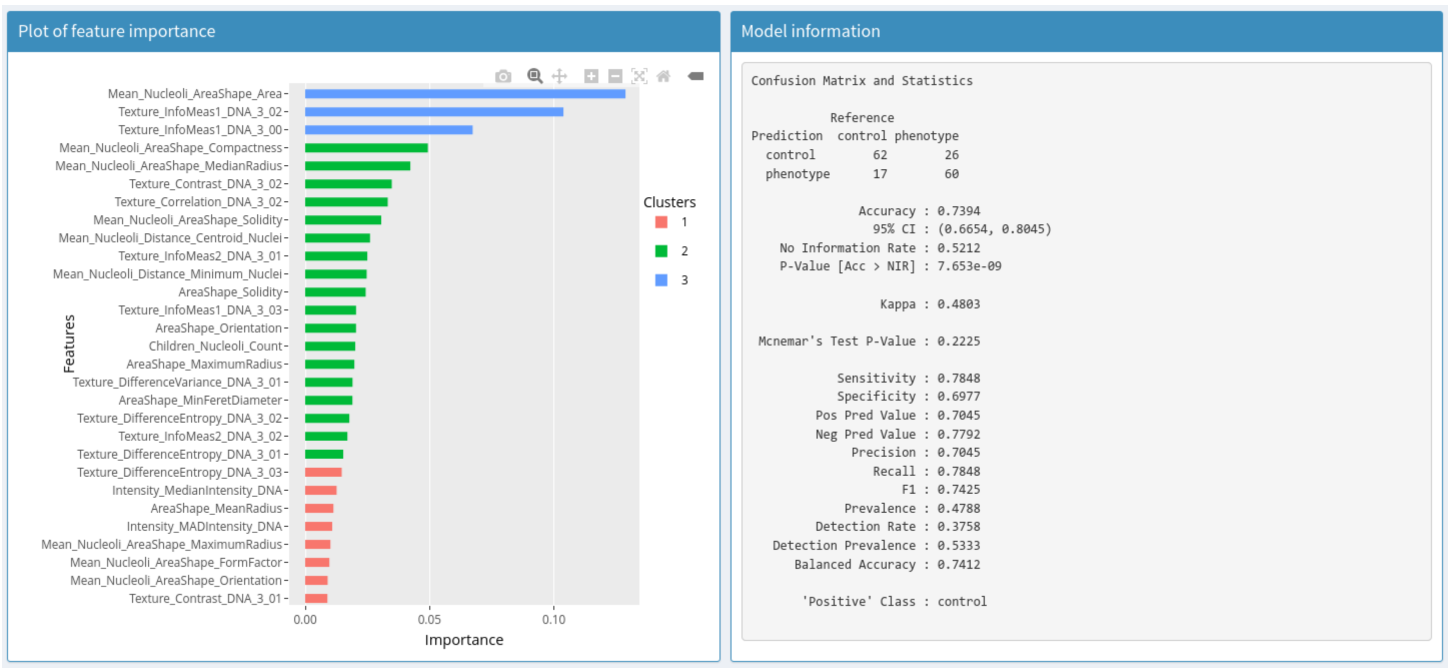
Location_Center_X	Location_Center_Y	Z.T	File
463.66	707.30	1.00	idr0002-chrCondensation-t170/plate1_2_006_[Well_92,_Field_1_(Spo

Figure 3

A



B


Plot of feature importance
Mean_Nucleoli_AreaShape_Area
Texture_InfoMeas1_DNA_3_02
Texture_InfoMeas1_DNA_3_00
Mean_Nucleoli_AreaShape_Compactness
Mean_Nucleoli_AreaShape_MedianRadius
Texture_Contrast_DNA_3_02
Texture_Correlation_DNA_3_02
Mean_Nucleoli_AreaShape_Solidity
Mean_Nucleoli_Distance_Centroid_Nuclei
Texture_InfoMeas2_DNA_3_01
Mean_Nucleoli_Distance_Minimum_Nuclei
AreaShape_Solidity
Texture_InfoMeas1_DNA_3_03
AreaShape_Orientation
Children_Nucleoli_Count
AreaShape_MaximumRadius
Texture_DifferenceVariance_DNA_3_01
AreaShape_MinFeretDiameter
Texture_DifferenceEntropy_DNA_3_02
Texture_InfoMeas2_DNA_3_02
Texture_DifferenceEntropy_DNA_3_01
Texture_DifferenceEntropy_DNA_3_03
Intensity_MedianIntensity_DNA
AreaShape_MeanRadius
Intensity_MADIntensity_DNA
Mean_Nucleoli_AreaShape_MaximumRadius
Mean_Nucleoli_AreaShape_FormFactor
Mean_Nucleoli_AreaShape_Orientation
Texture_Contrast_DNA_3_01
Features
0.00
0.05
0.10
Importance
Clusters
1
2
3
Model information

```
Confusion Matrix and Statistics

          Reference
Prediction  control phenotype
  control        62        26
  phenotype      17        60

               Accuracy : 0.7394
                 95% CI : (0.6654, 0.8045)
    No Information Rate : 0.5212
    P-Value [Acc > NIR] : 7.653e-09

                  Kappa : 0.4803

 Mcnemar's Test P-Value : 0.2225

            Sensitivity : 0.7848
            Specificity : 0.6977
         Pos Pred Value : 0.7045
         Neg Pred Value : 0.7792
              Precision : 0.7045
                 Recall : 0.7848
                     F1 : 0.7425
             Prevalence : 0.4788
         Detection Rate : 0.3758
   Detection Prevalence : 0.5333
      Balanced Accuracy : 0.7412

       'Positive' Class : control
```