System configuration:

All analysis/benchmarking was performed on a Dell PowerEdge R740xd server running Windows Server 2016 standard edition and equipped with: dual 3.3 Ghz Xeon Gold 6246 CPUs [Intel], 192 Gb DDR5 RAM, and a NVIDIA Tesla V100 32Gb purchased from Dell EMC using funds from the CPRIT CFSA RP150578. Scripts developed using Pipeline Pilot require a paid license that may be purchased from: <https://www.3ds.com/products-services/biovia/products/data-science/pipeline-pilot/>. All scripts have been tested on a machine running Windows 2016, no other operating systems have been tested. DNN feature embeddings have been optimized to run on a PC equipped with a GPU, however, this is not required.

In order to replicate the results of this study, code must be run in the following order:

1. **Project TL stack**
   1. Description – Automates and parallelizes ImageJ macro script to pre-process and project transmitted light Z-stack
   2. Software requirements:
      1. Pipeline Pilot 2018 server/Pro client
      2. Image J/Fiji (Downloaded at: https://imagej.nih.gov/ij/download.html) ImageJ must be integrated with Pipeline Pilot, see PLP/ImageJ installation instructions upon purchase.
      3. (ImageJ Plugin) Stack focuser (Download at: https://imagej.nih.gov/ij/plugins/stack-focuser.html)
   3. Runtime instructions
      1. Open Pipeline Pilot and load “1) Project TL stack.ppxml”
      2. Click on “Directory” and select folder containing the raw transmitted light images from an ImageXpress microconfocal
      3. Click “Run”
      4. Output: Folder with “\_Proj” suffix containing background corrected and projected images.
   4. Other:
      1. Tested on Windows Server 2016 and Windows 10
      2. CPU and RAM availability limit parallelization/speed. Requires > 4 gb RAM per thread.
2. **RESNET-embedding**
   1. Description – Converts images into numeric vectors using the RESNET-18 architecture that was pre-trained on the ImageNet dataset
   2. Software requirements:
      1. Anaconda with Python 3.8.X
      2. (Python package) PyTorch 1.7.1
   3. Run time instructions
      1. Open “2) RESNET-embedding.ipynb” using a jupyter notebook.
      2. Use the user input section to provide the path to the folder containing tif images
      3. Run “Guts” to load all functions, download pre-trained RESNET and perform image embedding.
   4. Output: DeepHTS.csv = Each row represents a single image, columns refer to index values of the RESNET-18 output feature vector.
   5. Other:
      1. Tested on Windows Server 2016 and Windows 10
      2. Typical run time: 1-3 minutes/384-well plate
3. **Feature mapping (PCA and kNN)**
   1. Description – Merges metadata (platemaps and drug info) then performs PCA analysis and kNN clustering using the output of the RESNET-embedding.
   2. Software requirements:
      1. Pipeline Pilot 2018 server/Pro Client or higher
      2. R 3.4+, must be configured to run with pipeline pilot according to the Pipeline Pilot installation instructions.
      3. (R library) caret
      4. (R library) e1071
   3. Runtime instructions
      1. Open Pipeline Pilot and load “3) Feature mapping (PCA and kNN).ppxml”
      2. Click on “Plate metadata” and select the plate metadata file. See the included file for example.
         1. Required columns in “Plate metadata.xlsx”
            1. Raw\_data = Relative file path to raw data directory
            2. Projections = Relative file path to projected images from “1) Projected TL stacks” algorithm. This folder (or subfolders with in) must contain the DeepHTS.csv file from “2) RESNET-embedding”
            3. CCL\_ID = name of the Cell line
            4. Plate\_map = Relative path to the plate maps. File must contain the following columns:

Row = 384-well plate row position

Col = 384-well plate row position

Cmpd1 = Compound name

Conc1 = The molar concentration of the drug treatment

Func = Experimental function code for drug treatment

Pos = Positive control

Neg = Negative control (typically DMSO)

Media = Media with out cells

Cells = Cells with media, no vehicle control

Sing = Single agent

DRC = Control response curve

* + - * 1. Assay\_batch = To group plates that are run in the same day/cell passage
        2. Group = Larger grouping that should be considered for modeling the cumulative dataset.
    1. Assumed image naming convention: <Channel>\_<Row>\_<Col>\_<Field>.tif
    2. Set “Group to run” to either primary or validation.
    3. Click “Run”
  1. Output: HTML report containing: PCA analysis, kNN quality control and class probability scores, SVR fitted dose response curves.
  2. Other:
     1. Tested on Windows Server 2016 and Windows 10
     2. Typical run time: 3-5 minutes