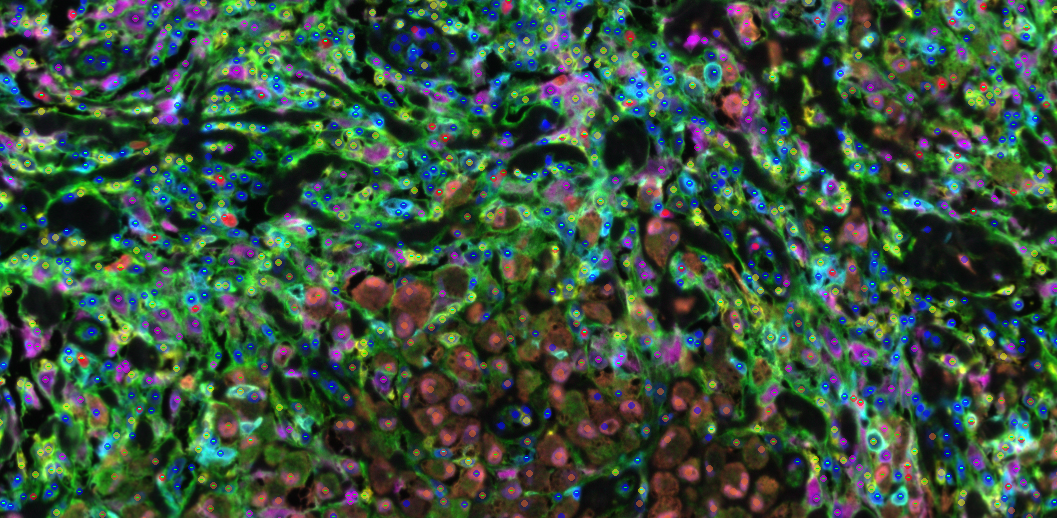
2019



Merge a Single Sample (MaSS)

Authors:

Benjamin Green1, Sneha Berry2, Elizabeth Engle2, Nicolas Giraldo3, Peter Nguyen2, Tricia Cottrell3, Janis Taube1,2,3, and Alex Szalay4

Affiliations:

Departments of 1Dermatology, 2Oncology, 3Pathology at Johns Hopkins University SOM, the Mark Center for Advanced Genomics and Imaging, the Sidney Kimmel Comprehensive Cancer Center, and the Bloomberg~Kimmel Institute for Cancer Immunotherapy at Johns Hopkins, Baltimore, MD, USA

Departments of 4Astronomy and Physics at Johns Hopkins University and IDIES, Baltimore, MD, USA

1. **Contents:**
2. Contents ………………………………………………………………………………………………………………………. pg. 2
3. Summary ………………………………………………………………………………………………………………………. pg. 2
4. Workflow Description …………………………………………………………………………………………………… pg. 3
5. BatchID Structure.…………………………………………………………………………………………………………. pg. 4
6. File Structure ………………………………………………………………………………………………………………… pg. 3
7. Installation/ running instructions ………………………………………………………………………………….. pg. 4
8. Output ………………………………………………………………………………………………………………………….. pg. 5
9. Image QA/QC ………………………………………………………………………………………………………………… pg. 6
10. **Summary:**

Merge a Single Sample (MaSS) is a program, written and compiled in MATLAB that facilitates the analysis of multiplex immunofluorescence imaging data. Specifically, it merges a set of binary phenotype classifications for individual markers created in inForm® Cell Analysis (Akoya Biosciences®) into a single coordinate system. In order to minimize over-segmentation and reconcile different cell segmentation algorithms the code is developed to satisfy the condition that only one cell is identified within 6 pixels of another cell call, a distance that is measured between cell centers. To reconcile conflicting classifications of the same cell, a hierarchical decision tree is used to determine which phenotypes will persist. The decision tree is embedded in the batch file along with information about the panel and pre-processing analysis. This document further details the steps involved in phenotype clean up and instructions on implementing the code

1. **Workflow Description:**

In order to run the program we first need to phenotype each marker in the multiplex panel using inForm® Cell Analysis separately. Label the two phenotypes as the antibody name, i.e. ‘CD8’, and ‘Other’. Next output the cell segmentation data in the format outlined below in section V. One of the main advantages of this workflow is the ability to use different segmentation algorithms for different antibody stains. Tips on how to configure an InForm® Cell Analysis project for compatibility with the MaSS program are defined at the end of Section VI and, in more detail, the accompanying protocol document ‘inForm Phenotype (Multipass).docx’. The next step would be to create a BatchID file as defined in section IV. This file will help define settings for how the cells will be merged such as marker type, which segmentation algorithm an antibody belongs to, which Opal was assigned to which antibody, and which cells can co-express.

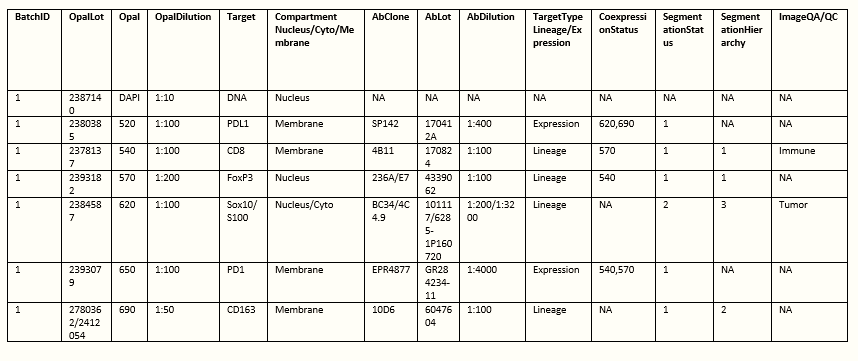
The following part of this document will be used to define the logic steps this program takes to merge the cell data into a single coordinate system. First, we define lineage markers and expression markers. Lineage markers, individually or in combination, are used to define specific cell types. Expression markers are those that can be expressed on multiple cell lineages at different levels. Cells without any lineage call in a radius of 6 pixels are classified as ‘Other’. We next define a primary segmentation which performs well on the majority of cell types featured in the panel. This is also where the ‘Other’ cell classification cell information, including x / y coordinates and intensity information, will come from. Secondary segmentation algorithms can be tailored for cells that do not segment well using the primary algorithm, i.e. larger tumor cells. For each of the lineage phenotypes we then check for cells that are within 6 pixels of each other and are of the same type. For these combinations the code removes the cells with the lowest total expression marker intensity. Cells may be designated to predominately express a specific expression marker and priority is placed on that expression marker intensity when removing cells.

Next the code resolves cells with different lineage phenotypes which are too close together using the segmentation hierarchy, defined in the batch tables, and corrects for acceptable co-expressions. In this hierarchy each lineage is ranked based off of the following multiple factors.

1. Uniformity of staining: we understand that some of these cell markers have a more uniform and believable staining pattern
2. Uniformity of cell type: we understand that some cells have patterns that are easier to distinguish and thus we should have a better confidence in the phenotype of these cells
3. Shape of cell type: we understand that some cells have different shapes that make them easier or harder for segmentation and classification algorithms to correctly identify
4. Importance of cell type in analysis; we understand that some cells are more or less biologically significant, thus we should make better attempts to capture these cell types

In order to further improve segmentation we remove `Other` cells with cell centers inside inForm defined cell membranes which are not in the primary segmentation, i.e. the secondary segmentation membranes. At this point the code reassigns unique CellIDs for each cell in the table.

Finally, the code assigns positive expression marker phenotypes to the final cell objects their cell centers are contained in. Expression marker phenotypes with cell coordinates inside two cell membranes are assigned to the closest lineage cell, based on cell centers.

1. **BatchID Structure**
   1. The BatchID files were designed to keep track of panel information such as reagants, concentration, and antibodies used. In order to make this program panel agnostic this spreadsheet is also used to provide input for the code. Specifically, the code uses the columns 3 ,5, 6, 10, 11, 12, 13, and 14.
   2. The BatchID tables should be the following:
   3. The first half of the spreadsheet describes the panel reagents. And is used to track reagent changes. The left half describes how the markers will be analyzed and merged to a single coordinate system where only one phenotype call exists for each cell; mark fields that do not have a designation as NA
   4. The ‘Target’ name should be used to name the opals in inForm, should be the name of the positive phenotype for each marker, and should be the name of the folder for each of the separate inForm outputs. The only exception is for the tumor marker, designated in the last column. For this marker, you should use ‘Tumor’ to designate the output folder, you may optionally also use ‘Tumor’ when designating that antibody in inForm either for the phenotype or for naming the opals.
   5. The last 6 columns allow the user to change the settings of the inform merge code
   6. TargetType Lineage/Expression
      1. This should be marked as either “Lineage” or “Expression”
      2. Lineage
         1. Markers that define the type of cell (ie Macrophage, TCell, Tumor, Treg)
      3. Expression
         1. Markers that can be expressed at different levels on many given cell types (PD1, PDL1) and will not affect the phenotype call of other cells
         2. All output will still be produced for each cell regardless of the opal designation but
   7. The CoexpressionStatus is **different for Expression and Lineage**
      1. for Expression markers input the opal dyes of the Lineage marker(s) it predominately co-expresses with
         1. in the above example PD1(650) co-expresses predominantly with CD8 (540) and FoxP3 (570), add “540,570” to this input for PD1
      2. for Lineage markers input the opal dyes of other Lineage marker(s) that will be allowed to co-express with each other
      3. in the above example CD8 (540) - FoxP3 (570) cells are accepted; so in the CD8 location add “570” for FoxP3 and in the FoxP3 location add “540” for CD8
   8. SegmentationStatus
      1. This is a numeric value; 1-X for the different types of segmentation that may exist, each marker with the same number should be phenotyped and processed with the same cell segmentation algorithm in inform
      2. i.e. in this panel all markers have the same segmentation (1) except Tumor (2)
      3. The primary segmentation (1) should be the more reliable algorithm and usually correspond to smaller cells
      4. “Other” cells will be defined by the primary segmentation (1)
   9. SegmentationHierarchy
      1. **For Lineage markers only,** create an order of phenotypes/ segmentation you believe will be most accurate – this is primarily based off of cell morphology
      2. Use NA for expression markers
      3. The code will remove cells according to this column
         1. for example: A cell given a positive phenotype for CD8 in the CD8 algorithm and positive for CD163 in the CD163 algorithm; if CD8 (1) is ranked higher than CD163 (2) then the CD163 designation will be removed
      4. Only cells that are allowed co-expression should have the same number. If two cells co-express with the same cell but not each other (CD4-FoxP3 and CD8-FoxP3); use the higher number for one of the two cells (CD4 or CD8) and the double co-expressed cell (FoxP3). Use a lower number for the other cell (CD4 or CD8). In the example with CD8-CD4-FoxP3 we usually use the numbers 1-2-1. CD4-FoxP3 will still be found based off of co-expression status but CD8-FoxP3 cells will take precedence.
   10. ImageQA/QC
       1. This column allows a user to set 2 different conditions of the batch
          1. The user can set one cell type as the ‘Immune’ cell
             1. There should only be one of these designations for the MaSS code to work
             2. This is the cell type that will define hotspots

For this version hotspots are defined as fields with the maximum CD8 cells

* + - 1. The user can set one cell type as ‘Tumor’
         1. This is an optional field and will allow the code to narrow down the ImageQA/QC to fields with more than 60 of this cell type
         2. There can be only one cell of this type
    1. Only set one marker as ‘Immune’ and only one marker as ‘Tumor’

1. **File Structure:**

The code relies on a data organization format detailed below:

1. set up the file structure as follows:
   1. \*DIR\ inform\_data\Phenotyped
      1. add a folder for each Antibody (ABx) in the panel
         1. ex:
            1. CD8
            2. CD163
            3. FoxP3
            4. Tumor
            5. PD1
            6. PDL1
         2. The antibody names here should correspond to those names used in the BatchID table. The only exception is the ‘Tumor’ marker which, if designated in column 14 in the BatchID table, should be label ‘Tumor’ here.
         3. Note: these names are all case sensitive
            1. If the folder names do not correspond to the Target names the code will produce an error to check the inform files
      2. in each folder export the \*\_cell\_seg\_data.txt files that inForm Cell Analysis® outputs for a phenotype analysis
      3. in the lowest numeric Opal of each segmentation type (described in the BatchID section) also export the \*\_binary\_seg\_maps.tif which inForm Cell Analysis® outputs for a phenotype analysis
         1. be sure these files have four layers

--- 1. Tissue Segmentation

--- 2. Nuclear Segmentation

--- 3. Cytoplasmic Segmentation

--- 4. Membrane Segmentation

* 1. \*DIR\MXX\inform\_date\Component\_Tiffs
     1. add all the \*\_component\_data.tiff images inForm Cell Analysis ® exports for each field analyzed

1. **Installation/ running instructions:**
2. install the application on the desired computer by opening the distributed file ‘MaSS Installer.exe’ and following the onscreen prompts. MATLAB does not need to be installed for the software to work.
   1. The installer will download and install a version of MATLAB runtime if it is not already installed on the computer. Note that the code will run off of whatever drive the runtime is installed on. Furthermore, if the runtime is installed on the wrong drive, using windows, uninstall the application and MATLAB runtime instance, then reinstall
3. Once installed, pass the following call to the cmd prompt:

CALL "C:\Program Files\Astropath\MaSS\application \MaSS.exe" "**\*DIR** \inform\_data\Phenotyped" "MXX" “**\*DIR** \BatchID\_XX.xlsx”

Replacing \***DIR** with the corresponding paths, MXX with the sample name, and BatchID\_XX with the name of the batch file. Unless changed during installation the path to the executable will be ‘C:\Program Files\Astropath\MaSS\application’. If the installation path is different, this should also be changed.

Once started. The code will locate the BatchID file and use this file to generate a data structure (called Markers) filled with information about the panel. After this table is populated the code will generate a resulting .csv for every image.

1. Tips for defining file structure, generating files, and creating projects
   1. Be sure to name the opals on the prepare tab in inForm Cell Analysis®.
      1. When naming the opals, use the same names as indicated in the file structure and the ‘Target’ column of the BatchID table.
      2. The labels in inForm will not be case sensitive
      3. For the Tumor marker, indicated by column 14, use the designation ‘Tumor’ instead of the full name, this is only required when creating the output folders, but may be useful in all aspects of this analysis
      4. Refrain from using any illegal characters like ‘\’ or ‘-‘
   2. When creating the tissue segmentation, include at least two categories labeling one as ‘NonTissue’
   3. When creating the cell segmentation algorithm be sure to check the boxes for ‘Membrane’, ‘Nucleus’, and ‘Cytoplasm’
   4. For the membrane segmentation outputs be sure that there are 4 layers
      1. \*if inForm Cell Analysis® is not exporting all layers to binary segmentation\*

--- open the project or algorithm in inForm Cell Analysis®

--- add and process an image to the export tab

--- click all segmentation layers to be visible and save the algorithm again, then use this algorithm to export the phenotype analysis

1. **Output:**

The code outputs two types of tables into a \*DIR\ inform\_data\Phenotyped\Results folder which is created upon startup. These outputs are separated into 2 subfolders:

1. \*Results\Tables contains the ‘\*\_cleaned\_phenotype\_table.csv’ & MaSSLog.txt
   1. The MaSSLog.txt details:
      1. the number of fields correctly processed by inform
      2. the number of merged files
      3. and the number of files printed which will be used by the QA\QC code
      4. As well as the criteria for the fields printed
         1. This will depend on user output and is detailed in the BatchID section as well as the ImageQA\QC section
   2. The code creates a ‘\*\_cleaned\_phenotype\_table.csv’ for each image processed. This table contains 62 columns:
      1. CellID – a unique cell id for each cell
      2. SlideID – the slide name
      3. fx & fy – x and y pixel coordinates of the field centers
      4. CellNum – the inForm Cell Analysis® unique cell id ( which may be rendered non-unique if multiple segmentation algorithms are used)
      5. Phenotype – cell lineage classification string corresponding to Lineage ‘TargetType’ ( described in the BatchID section)
         1. Each cell will have only one phenotype lineage unless it is included in a ‘CoexpressionStatus’ lineage pair in the BatchID table (described further in the BatchID section)
         2. In this case the phenotype will be presented as a string of ‘highest opal antibody’’lowest opal antibody’
            1. Ex:

CD8 – 540; FoxP3 – 570

‘FoxP3CD8’

* + 1. CellXPos & CellYPos – cell x and y pixel coordinates relative to the image
    2. EntireCellArea – Area of the cell in pixels
    3. Next are the intensity columns for each cell:
       1. For each opal there will be columns named as follows – where XXX is the opal number DAPI, 520, 540, etc.
          1. MeanNucleusXXX
          2. MeanMembraneXXX
          3. MeanEntireCellXXX
          4. MeanCytoplasmXXX
          5. TotalNucleusXXX
          6. TotalMembraneXXX
          7. TotalEntireCellXXX
          8. \*\*note TotalCytoplasmXXX is missing\*\*
       2. The columns are ordered by category and then opal such that all MeanNucleusXXX columns come before MeanMembraneXXX columns and so on
       3. These columns may contain NULL values but are otherwise float32
    4. Finally there are two column sets for expression marker status
       1. The first column set is for a binary classification:
          1. 1 positive & 0 negative
          2. Each column is named LOWER(ABx)

Ex. ‘pd1’

* + - * 1. The columns are in numeric opal order
      1. The second columns set is the confidence value from Inform Cell Analysis® for the expression marker phenotype
         1. Each of these columns is named LOWER(ABx)probability

Ex. ‘pd1probability’

\*\*note: this name is a misnomer as it is not really a probability measure, it is a confidence measure from the logistic regression done by inForm’s cell classifier\*\*

* + - 1. All columns from column set 1 come before column set 2
         1. Ex. [‘pdl1’, ’pd1’, ’pdl1probability’, ‘pd1probability’];

1. ‘\*\Results\tmp\_inform\_data’ contains .mat files for the images that meet the image QA\QC criteria detailed in the Image QA\QC and BatchID sections of the document
   1. These .mat files are deleted at the end of the Image QA\QC protocol; they essentially contain a copy of the ‘\*\_cleaned\_phenotype\_table.csv’ in an easily accessible MATLAB format
2. **Image QA / QC** 
   1. Description/ running instructions

In order to assess the performance of the cell phenotype algorithms on a large quantity of images, an algorithm was developed to selectively sample images and create modified visual displays of those images. This algorithm is a secondary program which must be downloaded and installed separately, it is called CreateImageQAQC. This code must be run after the MaSS protocol as it relies on the tmp\_inform\_data directory the MaSS tool creates. The code is also relatively simple to run from a cmd prompt using the following:

CALL "C:\Program Files\Astropath\ CreateImageQAQC \application \ CreateImageQAQC.exe" "**\*DIR** \inform\_data\Phenotyped" "MXX" “**\*DIR** \BatchID\_XX.xlsx”

Again replacing \***DIR** with the corresponding paths, MXX with the sample name, and BatchID\_XX with the name of the batch file. Unless changed during installation the path to the executable will be ‘C:\Program Files\Astropath\MaSS\application’. If the installation path is different, this should also be changed.

* 1. Workflow Description:

The code first selects images for quality control based on user specifications. Only, images containing at least 60 tumor cells and images with at least 75 percent tissue coverage are considered for each sample. If a tumor cell designation is not included in the BatchID table, the tumor cell search criteria is removed. If less than 20 fields in the sample meet the search criteria, the minimum tissue coverage requirement is reduced by 6.25 percent. This is repeated until 20 fields meet the search criteria or the field tissue coverage search criteria decreases below 50 percent. If 20 fields or less than 20 fields are selected then performance testing is carried out on all fields. If more than 20 fields are selected, then the 20 highest designated ‘Immune’ cell density fields are selected for performance testing.

To assess assigned co-expression, pie charts are created to show cell proportions in each field, shown on the left. The first pie chart shows portions of all cells in the field, while the others display the proportion of cell lineages expressing each expression marker type. In this way, it is easy to see if expression patterns assigned correlate with current biological understanding of antibody performance. A heat map of log2 detected mean intensity in cells vs their assigned lineage phenotypes is also generated. When the phenotype algorithms are performing correctly high opal intensity should correlate with the marker it is labeling. For example, CD8 is labeled by Opal 540, if the phenotype is performing well there should be high intensity in the Opal 540 - CD8 rectangle.

While these metrics aid in performance assessment, visual inspection is the final verdict. To do this cell stamp mosaics are generated for each marker. In each mosaic, 50 cell stamps are randomly sampled from the image, 25 of which were centered on positive cells and 25 centered on negative cells. In these image stamps, the image segmentation in red as well as a grey scale of the component being detected is displayed. In order to assess the global context and performance of cells in each image, full size composite images are also generated. Full size composites are generated displaying each antibody separately as well as together, both with and without the segmentation overlaid on top.

* 1. Output Structure

The code creates three outputs into a QA\_QC subfolder under the \*DIR\MXX\inform\_data\Phenotyped\Results folder created in the MaSS protocol.

* + 1. Tables\_QA\_QC
       1. These tables are the same output as in DIR\MXX\inform\_data\Phenotyped\Results\Tables, but are placed here for convenience if further testing is desired
    2. ImageQA\_QCLog.txt
       1. This log file details the number of hotspot fields chosen, as well as time stamps for the image output, the figure output, and completion time of the program.
    3. Phenotype
       1. Under this folder will be a subfolder called All\_Markers
          1. In this folder there are three types of image output, designated by the following strings after the image names

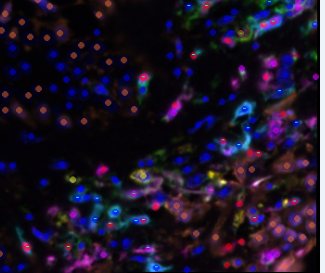
‘\_cleaned\_phenotype\_image’

This image shows the full image with all component layers

The color-marker pairs are indicated in the bottom left hand corner of the image

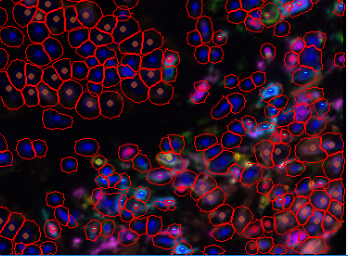
Assigned phenotypes are indicated by the dots overlaid on the sample

Lineage markers assignment is designated by the color of the circle while expression markers are by the horizontal strips of each cell

Coexpressing lineages are designated by half dots where the top half indicates one marker and the bottom half indicates the other

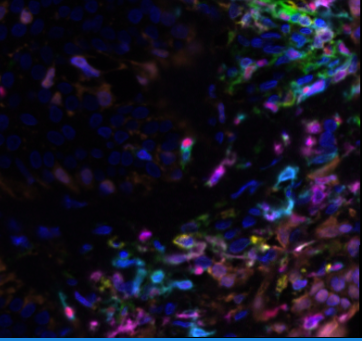
‘\_cleaned\_phenotype\_w\_seg’

This is image is the same as the image in ‘b’ except that is has the segmentation overlaid on top





‘\_composite\_image’

This image displays the full image with all component layers but without the dots indicating phenotype

* + - 1. There will also be a sub folder for each lineage and expression marker designated
         1. This includes a folder for any combination lineage markers designated in the ‘CoexpressionStatus’ from the BatchID table
         2. In these folders output is shown to assess the performance of that marker
         3. There are four types of images, designated by the following strings after the image names

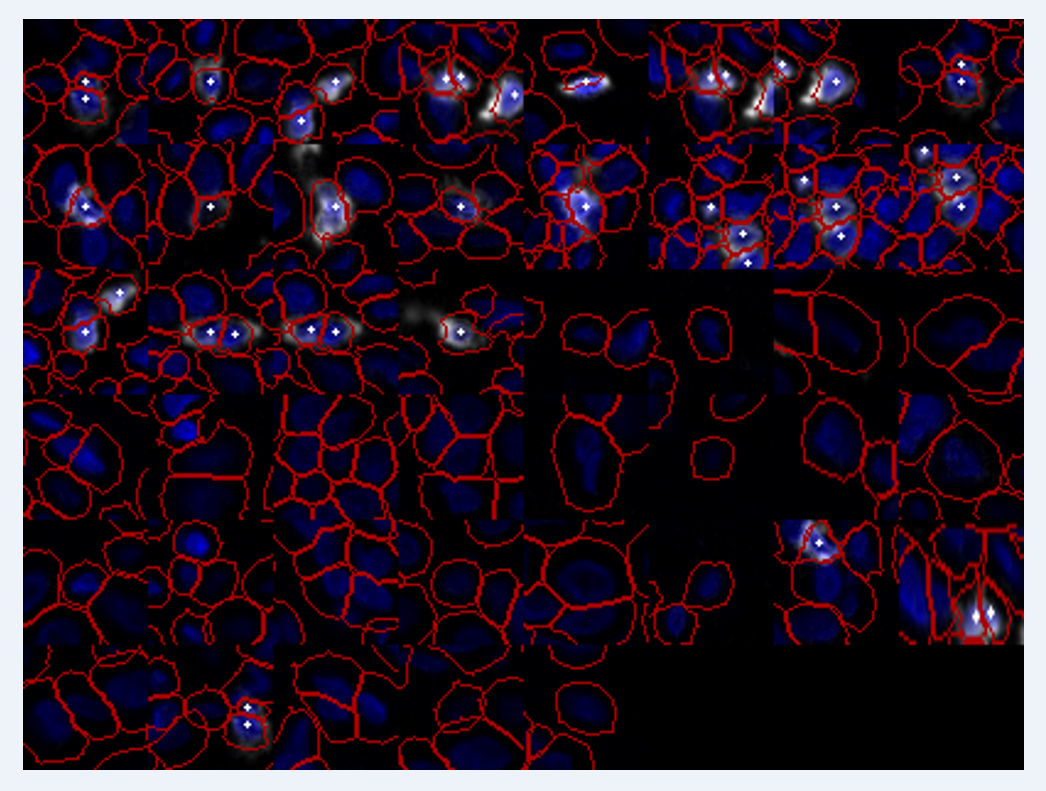
‘\_cell\_stamp\_mosaics\_pos\_neg’

These are mosaics of cell stamps, which include the cell segmentation, DAPI, and a grey scale of the component expression overlaid

Up to 25 positive cells for the selected marker are randomly sampled with 25 negative cells

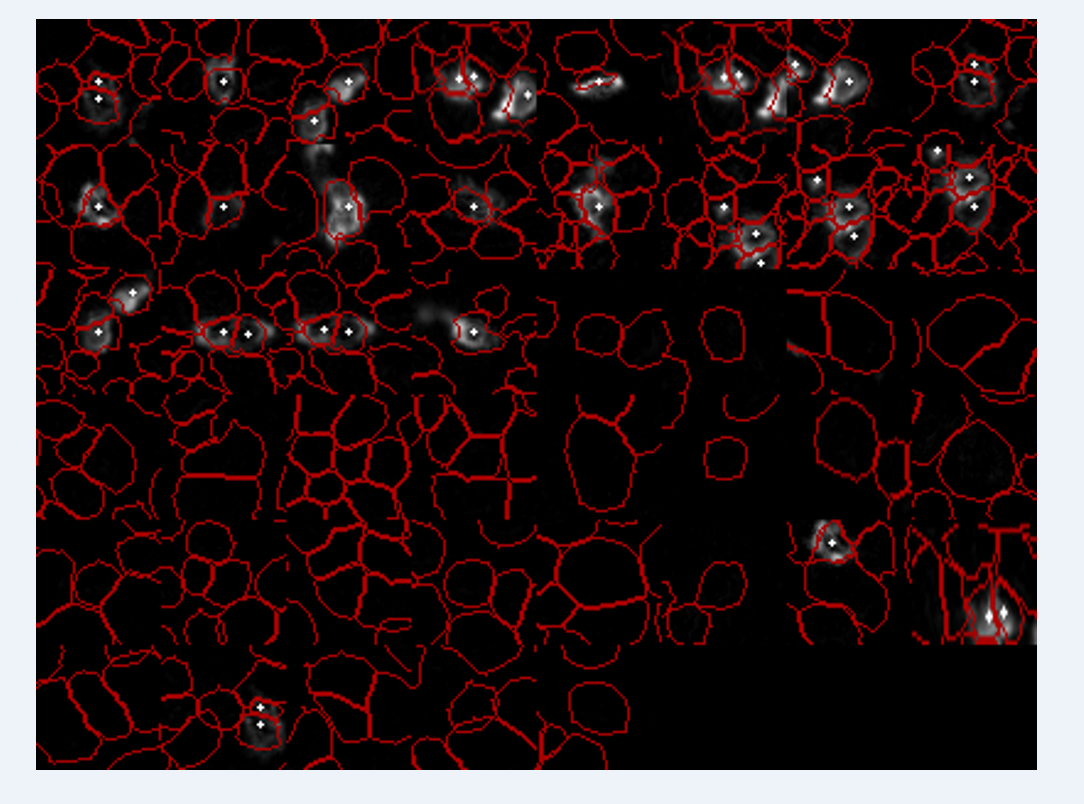
If more than 25 cells of this type do not exist in the image, all cells are then sampled

A 50x50 pixel board has been cut out around the cell so that the cell of interest in the middle of the cell stamp

The white crosses indicate positive cells for the marker of interest

‘\_cell\_stamp\_mosaics\_pos\_neg\_no\_dapi’

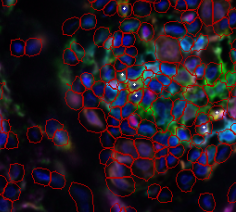
These mosaics are the same as previously described with DAPI removed to increase visibility

The white crosses indicate positive cells for the marker of interest

‘\_full\_color\_expression\_image’

Full image with all component layers and segmentation overlaid

The color-marker pairs are indicated in the bottom left hand corner of the image

The white crosses indicate positive cells for the marker of interest

‘\_single\_color\_expression\_image’

Full image of the component layer of the marker of interest in grey scale, DAPI, and segmentation overlaid

The white crosses indicate positive cells for the marker of interest