Cytokit: A single-cell analysis toolkit for high dimensional fluorescent microscopy imaging

Eric Czech*, Bulent Arman Aksoy*, Pinar Aksoy*, Jeff Hammerbacher*
* Microbiology and Immunology Department at Medical University of South Carolina

Abstract

Cytokit is a collection of open source tools for quantifying and analyzing properties of individual cells in large fluorescent microscopy datasets that are often, but not necessarily, generated from multiplexed antibody labeling protocols over many fields of view or time periods. Cytokit offers *(i)* an end-to-end, GPU-accelerated image processing pipeline; (ii) efficient input/output (I/O) strategies for operations specific to high dimensional microscopy; and (iii) an interactive user interface for cross filtering of spatial, graphical, expression, and morphological cell properties within the 100+ GB image datasets common to multiplexed immunofluorescence.

Image processing operations supported in Cytokit are generally sourced from existing deep learning models or are at least in part adapted from open source packages to run in a single or multi-GPU environment. These operations include registration across imaging cycles (a critical step in multiplexed image analysis), deconvolution, point spread function (PSF) generation, image quality assessment, and cell/nuclei segmentation. The remaining tools offered facilitate visualization, image subset extraction with ImageJ compatibility, and iterative pipeline optimization.

Source code, documentation, and example data sets are freely available under the Apache License 2.0 at https://github.com/hammerlab/cytokit.

Introduction

Molecular profiling of cell culture and tissue samples traditionally relies on techniques that do not support a diverse panel of protein targets without disturbing important *in situ* characteristics of cells. Immunofluorescence imaging preserves these characteristics but is limited to a small number of expression measurements due to the need to avoid overlapping fluorophore emission spectra. This limitation can be overcome to an extent through repeated imaging of the same specimen over several cycles, however the incubation period necessary between cycles is often

hours or days and methods for removing markers from previous cycles can be detrimental to assay quality. By contrast, techniques like Mass Cytometry [1] and Multispectral Flow Cytometry [2] enable the measurement of more target compounds but provide little to no morphological or spatial information. Other methods such as Multiplexed Immunohistochemistry [3] and Multiplexed Ion Beam Imaging [4] overcome these limitations but require special appliances that are not compatible with standard or commercial microscopy platforms. For these reasons, analysis of data from multiplexed fluorescent labeling methods are appealing as they are economical, can be conducted with any fluorescent imaging platform, and rely on well documented immunostaining protocols.

Methods developed for multiplexed fluorescent labeling include Co-Detection by Indexing (CODEX) [5], DNA Exchange Imaging (DEI) [6], and t-CyCIF [7], the first two of which operate by labeling a sample one time with a large (possibly >100) number of antibodies conjugated to oligonucleotide sequences that act as temporary fluorophore binding sites. Fluorophores bound to complementary sequences are then introduced to enable the collection of a small number (typically 4) of fluorescent images between staining cycles. These procedures are capable of measuring the expression of tens or hundreds of different proteins but introduce several key computational challenges that make employing them difficult. The primary challenge is that incorporating the larger number of expression targets within 3D image volumes that encompass hundreds of thousands of cells, a sample size often necessary for studying less common cell types [8], leads to 100+ GB raw image sets including well over 10,000 images. As this data is often the product of commodity imaging platforms, substantial processing is necessary for analysis and the time associated with this processing becomes prohibitive for high dimensional acquisitions without GPU acceleration. With GPU acceleration however, we show how a CODEX experiment, shared by Goltsev et al., including 51k 16-bit images (129 GB), 54 expression markers, and ~70k individual cells can be aligned, deconvolved, and segmented for analysis in less than 90 minutes per workstation.

Here, we present and validate a library providing GPU accelerated implementations of image processing algorithms often required for, but not exclusive to, analyzing high dimensional immunofluorescence data. This library is open source, requires nothing more than an existing nvidia-docker installation to run, is adherent to best practices such as continuous integration and unit testing, and includes a novel user interface to help navigate the cellular characteristics measured by this next generation of fluorescent imaging technology.

Implementation

Cytokit consists of four major components, each of which will be discussed further in the following sections:

- **Processing** - Processing operations are applied to tiled images in parallel and in a predefined order to maximize ease of use as well as ensure that I/O for large image files is minimized;
- **Configuration** - Beginning from a template configuration, image operations and extractions can be enabled/parameterized through “deltas” to concisely define variations on experimental results;
- **Extraction** - Cytometric single cell data can be exported as FCS or CSV files and individual channels can be grouped into arbitrary subsets for extraction into ImageJ [9] compatible TIFF hyperstacks for custom analysis;
- **Visualization** - The Cytokit Explorer UI allows for Individual cells to be isolated based on phenotype and visualized within the entire field of view for an experiment, individual fields of view, or as single cells.

Processing

The imaging processing pipeline in Cytokit, based largely on the original CODEX Project, is designed to support datasets with the following dimensions:

- Tile - A single field of view
- X/Y/Z - 3D images captured at each tile location
- Channel - Images for different expression markers
- Cycle - Groups of channels (usually 3 or 4) captured between dye exchange cycles
- Region - Groups of tiles often collected as square grids

This seven dimensional structure, illustrated in **Figure 1**, is common in multiplexed imaging but dimensions of length 1 are also supported in nearly all cases. In other words, the smallest Cytokit-compatible experiment would consist of a single, grayscale 2D image with a height and width exceeding some minimum value (currently 88 pixels). All of the imaging processing steps applied to these datasets can be enabled or disabled based on the experiment configuration and are executed in a predetermined order. The individual steps themselves are implemented using existing Deep Learning models or are ported from other Java or Python libraries to run as GPU-accelerated TensorFlow [10] computational graphs. There are some exceptions to this such as image I/O and cell segmentation post-processing, but all of these are discussed below.

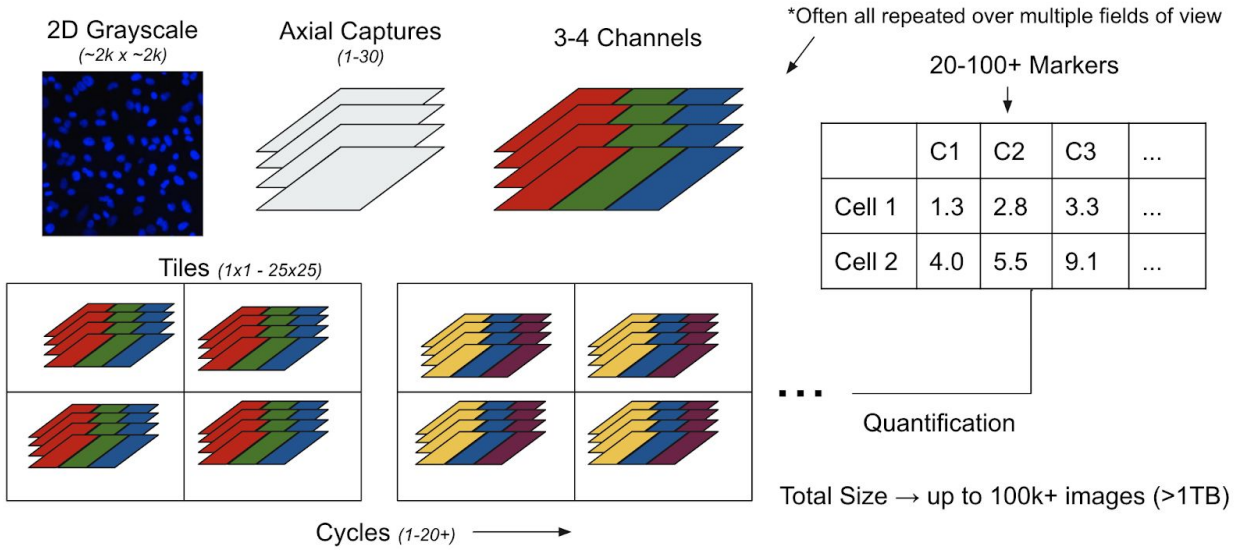

Figure 1: Seven dimensional multiplexed experiment structure illustrating how 3D single channel images are grouped as cycles, captured as tiles on a grid, and then potentially repeated over multiple fields of view before being quantified as two dimensional single-cell information.

Operations

As illustrated in **Figure 2**, the first step in the Cytokit pipeline involves loading images into memory in a way that is decoupled from all further processing steps. This is done using a separate thread, outside of any TensorFlow graphs, to assemble 2D grayscale images as 5 dimensional tile arrays that are then loaded onto a queue of configurable size (typically 1). This ensures that as processing for a single tile completes, there is no delay introduced by disk I/O before beginning processing on the next tile. This is an important feature for maximizing GPU utilization and reduces overall processing time by up to 20% in our experiments. The image files themselves are assumed to be 8 or 16-bit 2D grayscale images with file names containing region, tile, cycle, z-plane, and channel index numbers, in a configurable format, so that the array structure matches that of the experiment.

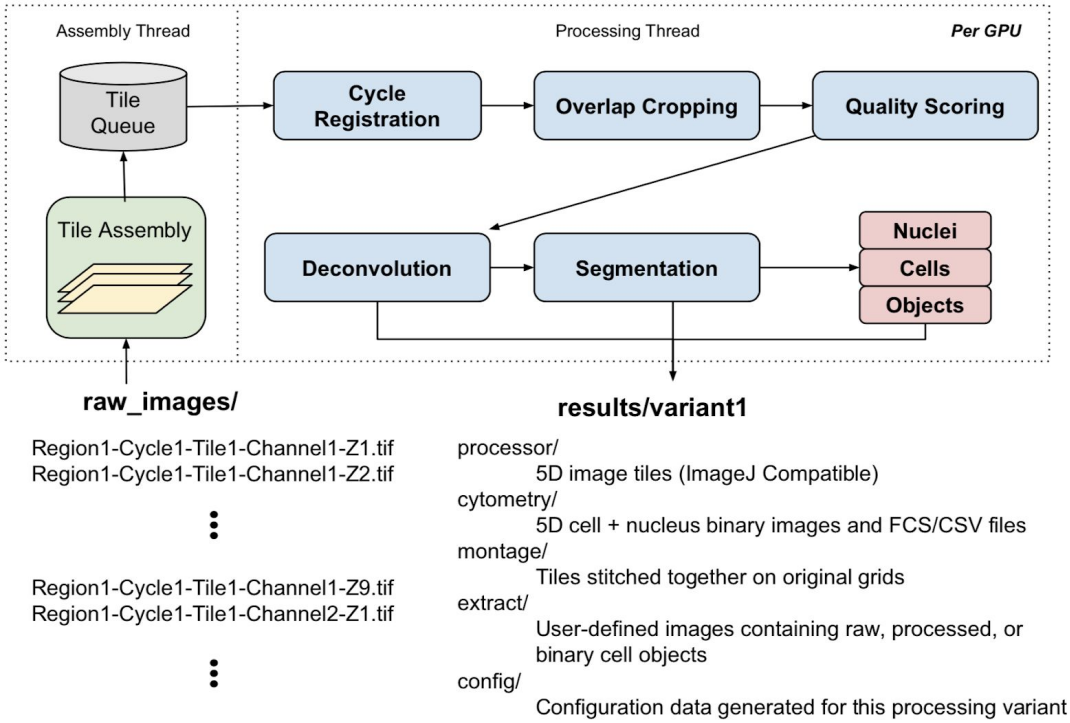

Figure 2*:* Processing pipeline overview with operations in original CODEX project replaced by GPU-accelerated equivalents, decoupled from tile assembly, and modified to support labeled object extraction as well as ad-hoc image stacks/montages

The remaining image manipulation operations include:

Cycle Registration - This operation serves as a way to align reference images across cycles and then apply the inferred translation to all other non-reference images. As is necessary in multiplexed imaging, a reference channel usually containing a nuclear stain must be collected in each cycle so that the sample drift occurring in the time elapsed between fluorophore exchanges can be compensated for in downstream analysis. In Cytokit, this operation is implemented as a port of the cross correlation algorithm [11] in scikit-image [12] `register_translation` to TensorFlow.

Image Quality Assessment - This is provided by the Microscope Image Quality [11,13] project, a TensorFlow classifier that allows for images to be scored based on their quality, and is used in Cytokit to select individual 2D images for visualization and/or quantification.

Deconvolution - 3D image deconvolution in Cytokit is provided by the Flowdec project, which is a direct port of the Richardson Lucy algorithm in the DeconvolutionLab2 [14] library to TensorFlow. Cytokit also includes support for automatically generating Point Spread Functions based on an experiment configuration through a fast Gibson-Lanni kernel approximation method [15].

Segmentation - Identification of cell nuclei is performed using the Deep Learning model [16] featured in CellProfiler [17]. The semantic segmentation provided by this model is then used to identify entire cell objects based on either a fixed radius outside the nucleus or, if available, a membrane stain channel used to create a threshold image severing as a watershed mask. Smaller objects within a cell can also be quantified as “spots”, a process which is again implemented using a threshold image created from any user-defined channel. All processing subsequent to semantic segmentation is implemented using scikit-image, SciPy [18], and OpenCV [19].

Configuration

Experimental variation in high throughput microscopy imaging often requires processing pipelines that are very tunable. While the intention in Cytokit is to employ algorithms that require this as little as possible, some configurability is unavoidable. The approach taken to address this problem involves two core capabilities -- iteration and evaluation. Much like any general hyperparameter optimization process, Cytokit attempts to make defining iterations and evaluating them as simple as possible so that tuning individual operations for individual images is much less common than being able to view the effects of parameter settings across an entire experiment. For example, a common process in our lab is to generate a template configuration that is then modified incrementally to produce several variants of an experiment for analysis. A simple version of this process is shown in **Figure 3** and demonstrates how the volume of data incorporated may be increased as appropriate parameter settings become clearer.

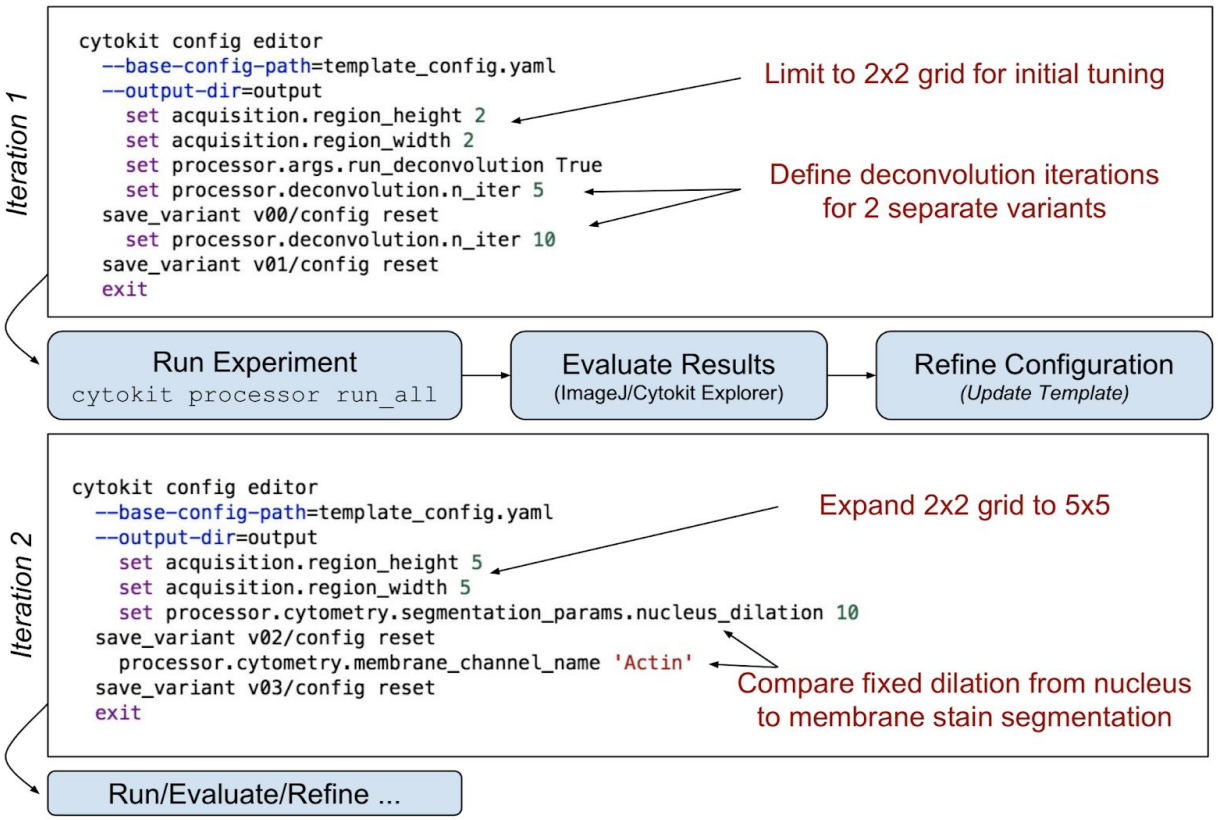

Figure 3: Example iterative pipeline optimization process with CLI commands used to continually refine and expand the scope of experiment processing for large raw image datasets

Template configurations, which may also simply be the sole configuration for an experiment, need to at least include the dimensions of the experiment (grid size, number of Z planes, image overlap, etc.), microscope parameters, and channel names. They may also contain definitions for any commands to be run as a way to ensure that all processing for an experiment is defined in one place. Practically speaking, this means that arguments to a command line interface (CLI) controlling processing, data extraction, or analysis may be specified in the experiment configuration or overridden at the command line. This often fosters reproducibility since it eliminates the need to keep track of how a pipeline was invoked while still making it possible to introduce small changes in behavior. A good example of this is defining channel subsets to extract for visualization as this can be done completely ad-hoc at a command line or defined in the template configuration if the extraction is commonly useful.

Extraction

Extracting data from high dimensional experiments can be challenging, particularly when slices of interest across those dimensions span image volumes that are too large for other visualization and analysis software. Extraction utilities in Cytokit make it possible to mix raw

image data with processed image results as well labeled object data (for cells/nuclei) and can be parameterized to operate on subsets of image grids or fields of view, lists of specific channel names, or z-plane subsets. Those extracted images can then also be stitched together such as in **Figure 4** where a 49 tile (7x7 grid) montage of 1008x1344 images was generated as single TIFF file (cropped to 8192x8192 pixels) containing marker expression levels and cell/nucleus boundary data.

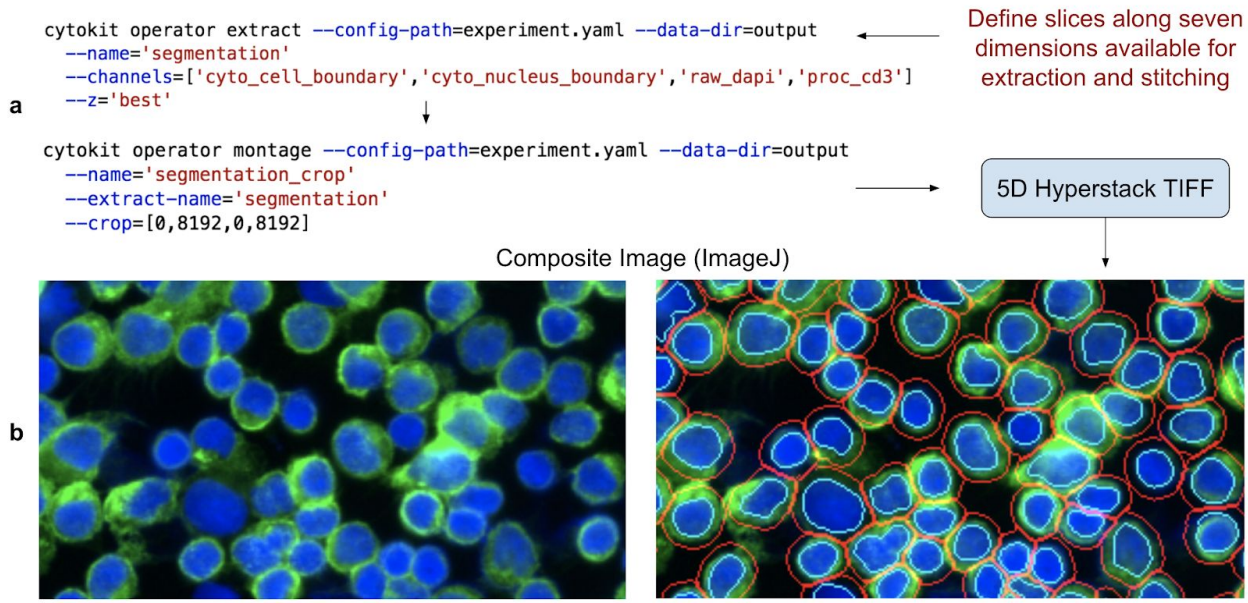

Figure 4: Example extraction and montage CLI commands. **(a)** CLI commands define slices as names (where relevant) or as lists and ranges of indexes to extract raw, processed, or object image data. **(b)** Resulting TIFF files are ImageJ compatible for blending and visualization, like the example shown with human T cells labeled as CD3 (green), DAPI (blue), nuclei boundaries (cyan), and cell boundaries (red).

In addition to image extraction utilities, Cytokit also offers single cell data as FCS or CSV files containing cell identifiers linking back to object images, location coordinates within the experiment, morphological properties (diameter, size, circularity, etc.), mean expression levels across entire cell or within nucleus alone, and graphical properties like identifiers of adjacent cells, number of adjacent cells, and size of contact boundary.

Visualization

One of the primary challenges in in-situ image cytometry is developing an understanding of the relationship between phenotypic and spatial properties of cells. Multiplexed imaging further complicates this process as the quantity of images produced both increases the likelihood that illumination artifacts exist along the spatial dimensions and often makes finding them manually infeasible. To assist in interrogating these relationships as well as build phenotypic profiles for

cell populations, an interactive Dash (by Plot.ly) application is provided that allows for gates applied to cytometric data to be projected onto images within an experiment. Shown in **Figure 5,** this application can be used to visualize individual cell/nucleus segmentations, highlighted as SVG overlays, within the context of a stitched grid view of an experiment as well as a single field of view. Additionally, single cell images can be extracted from an entire experiment with blended overlays of various expression channels (having user defined contrasts) as a means of ensuring that the characteristics of cells assumed to exist in any one population gated purely based on numerical information match expectations.

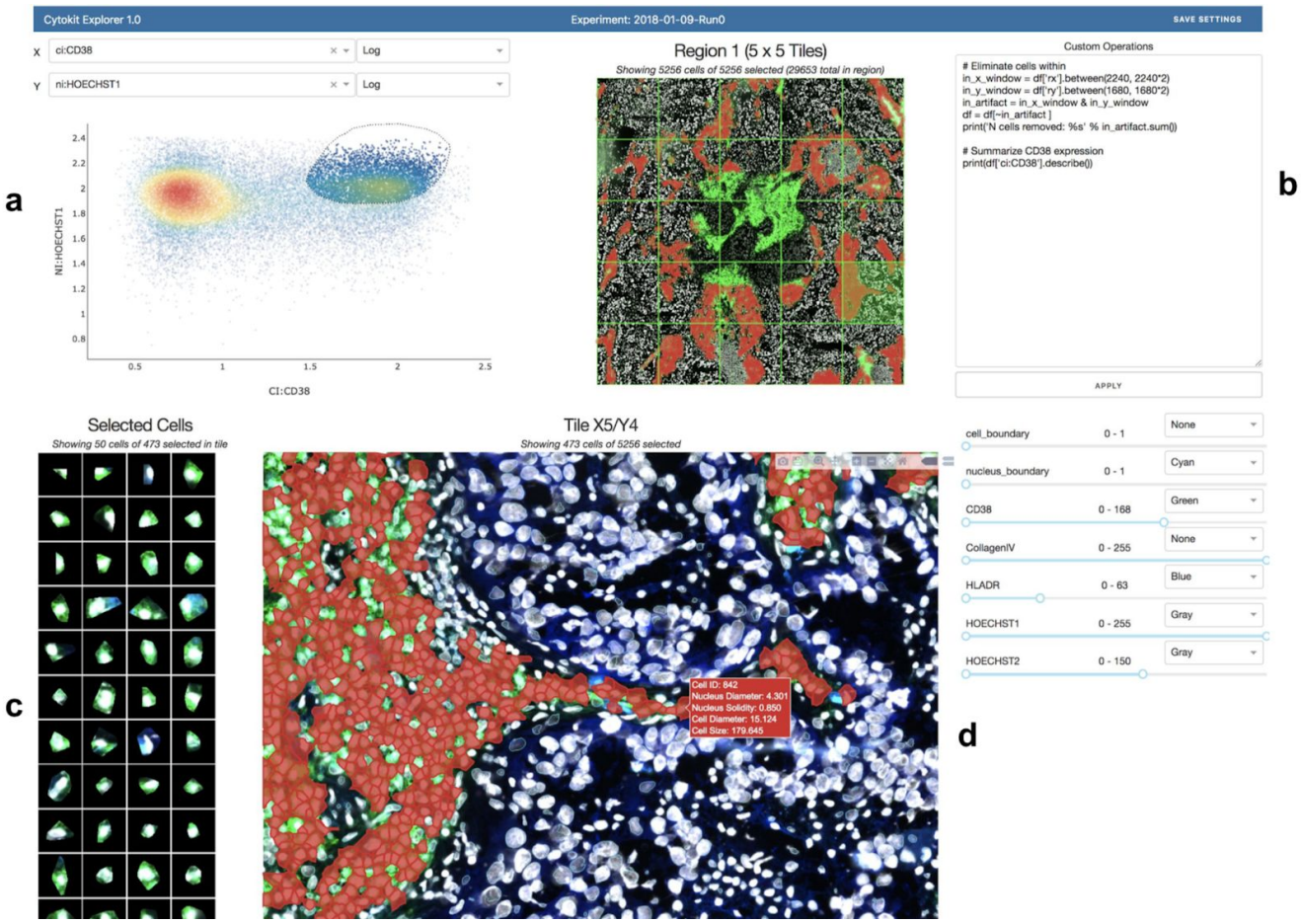

Figure 5: Cytokit Explorer screenshot (see screencast for animated version) showing a CODEX sample imaged at 20x. **(a)** 1D or 2D plots of expression, morphological, or graphical cell features support box and free-hand gating. **(b)** Custom filtering, to ignore a central photobleached region of cells in this case, or summarizations are applied immediately. **(c)** Single cell images match current gate and selected channel display settings and can also be buffered onto the page as tiles are selected, or across the entire image grid (not shown). **(d)** Gated cell population projected onto selected tile image with current display settings

Validation Results

Cellular Marker Profiling

To demonstrate the extraction and analysis features of Cytokit as well as validate the underlying image processing libraries, a series of traditional immunofluorescence experiments was first conducted on human primary T cells. The first of these, shown in **Figures 6** and **7**, comprised of primary human T cell samples stained with CODEX oligonucleotide-conjugated antibodies against human CD3, CD4, and CD8 (and a separate HOECHST stain). These slides were then imaged at 20X on a 1.9mm x 2.5mm grid of 25 images over an axial depth of 12.5 micrometers. The resulting 1008x1344x25 (height x width x depth) image volumes were then deconvolved, segmented, and quantified before being gated using Cytokit Explorer to isolate helper (CD4 positive) and cytotoxic (CD8 positive) subpopulations.

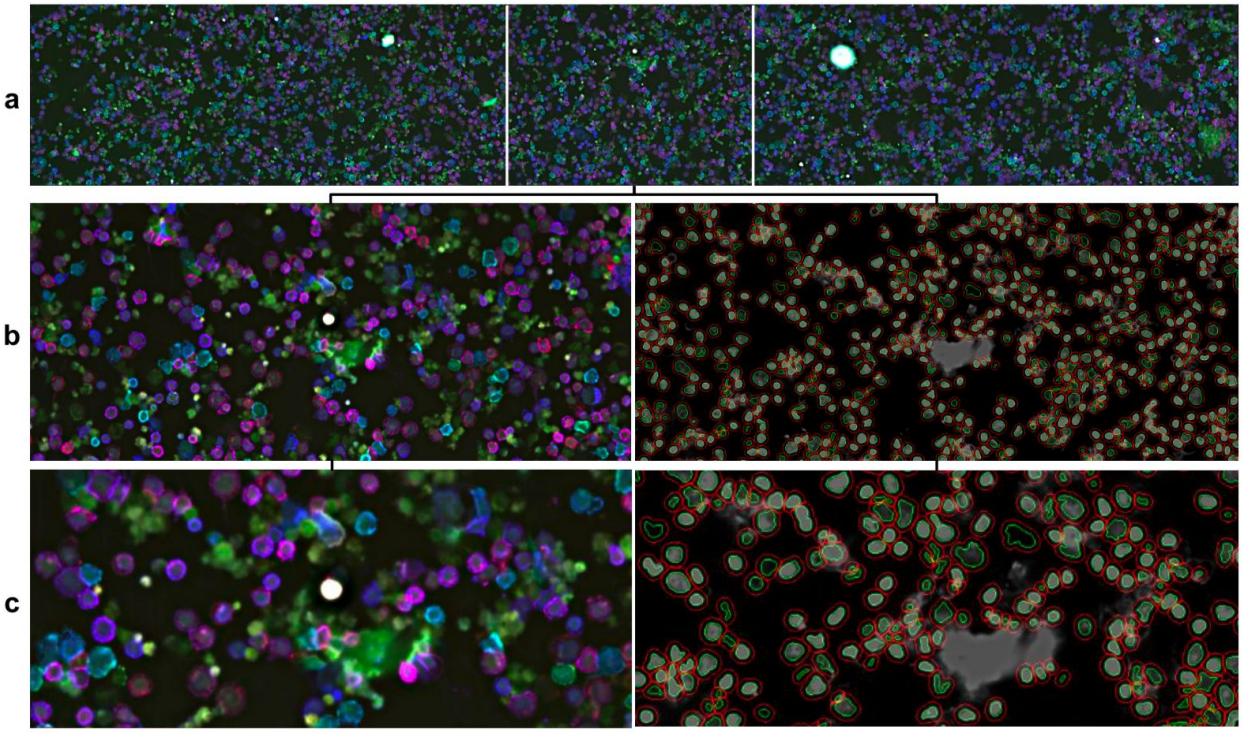

Figure 6: T cell CD3 (blue), CD4 (red), and CD8 (green) intensity. **(a)** First row of 5 images in 5x5 experiment grid. **(b)** Single tile image with corresponding cell and nucleus segmentation, where cells are defined as a fixed radius away from nucleus in the absence of a plasma membrane stain. **(c)** Center zoom on (b) showing co-expression of CD3 and CD4 (magenta) and CD3 and CD8 (cyan) as well as debris in DAPI channel.

While these CD4 and CD8 positive (i.e. CD4+ and CD8+) populations were easily resolved in this experiment (**Figure 7**), we found that dissociated cell samples like this are difficult to

prepare without non-trivial amounts of debris and diffuse nuclear staining, usually as a result of lysed cells that did not survive the centrifuge. This is visible in **Figure 6 (c)** where a minority of the nuclei segmentations resulting from the CellProfiler U-Net are fixed around roughly circular areas of greater DAPI intensity, albeit at low contrast. This invariance to contrast is generally very desirable, but it also demonstrates the importance of curation in image cytometry as artifacts like this can easily go undetected without a way to relate variations in inferred morphological or expression profiles for cells back to raw images.

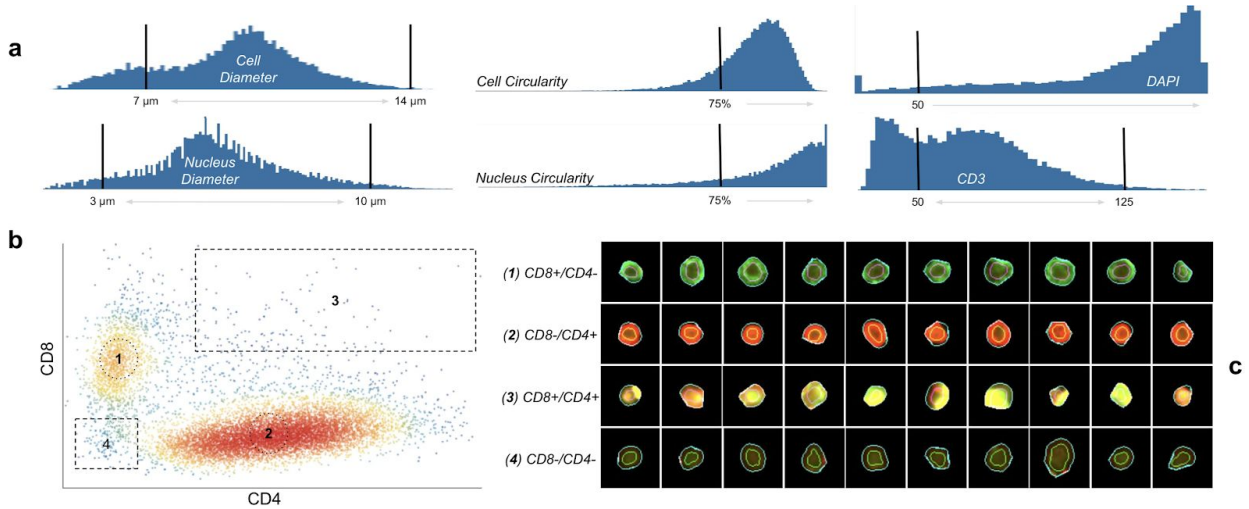

Figure 7: T cell gating workflow as annotated Explorer screenshots. **(a)** Morphological and intensity gates applied to isolate CD3+ cells. **(b)** Cytotoxic and helper cell subpopulations. **(c)** Individual cell images matched to subpopulations in (b).

A further investigation of the ability of this method to isolate CD4+CD8- and CD4-CD8+ cell populations was also conducted on experimental replicates and validated against flow cytometry based surface marker profiling. Shown in **Figure 8**, population proportions matched closely and verified that dissociated cells quantified in this way can produce results comparable to other methods; however, a tool like Explorer was necessary to establish appropriate morphological as well as spatial gates to ignore debris and portions of individual image tiles before this degree of parity could be attained.

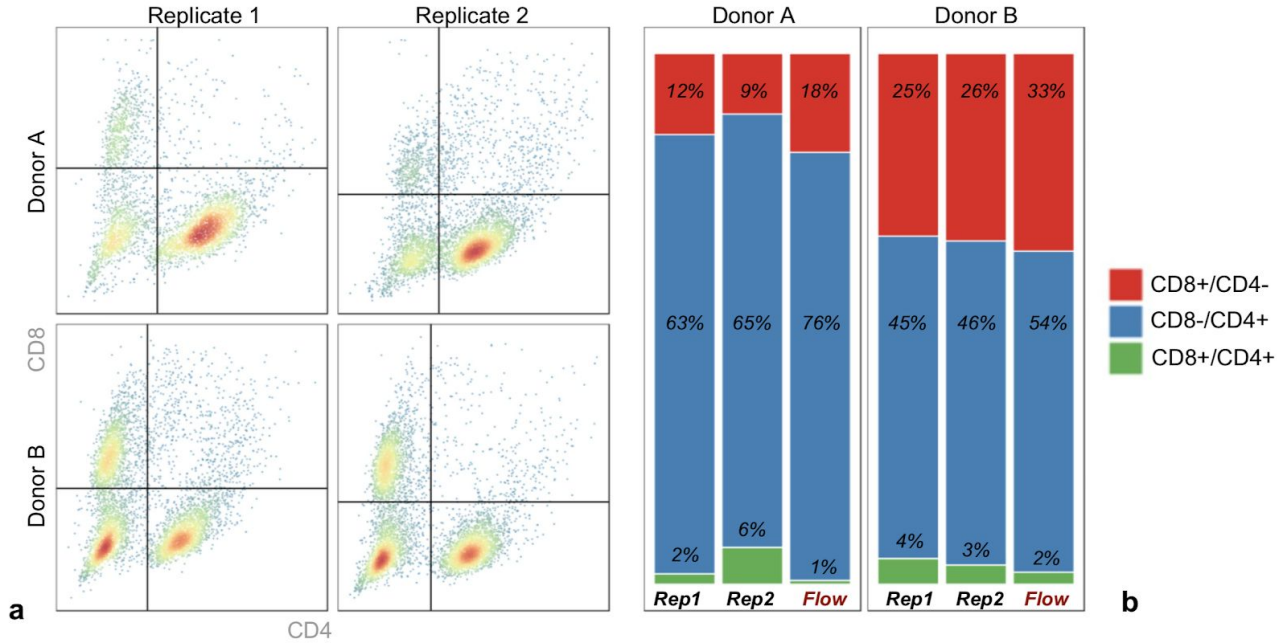

Figure 8: T cell population recovery comparison (source code). **(a)** CD4/CD8 gating results, as determined by automatic gating functions in OpenCyto [20], for imaging replicates over two donors. **(b)** Population proportions of both replicates compared to a single Flow Cytometry measurement for each donor.

Cell Size Estimation

A second validation was also conducted to attempt recovery of known morphology differences between unstimulated and activated T cell samples. Samples were stained with DAPI and Phalloidin-Fluor 594, the latter of which identifies cytoskeletal actin filaments used here to define cell boundaries. The Cytokit pipeline for processing both samples was configured to blur actin images before applying a threshold (Otsu) to create a binary image used as a watershed segmentation mask, with nuclei as seeds. As in the previous experiments, Explorer was used to find appropriate gates for singlet cell populations before attempting to compare cell diameter distributions (**Figure 9**).

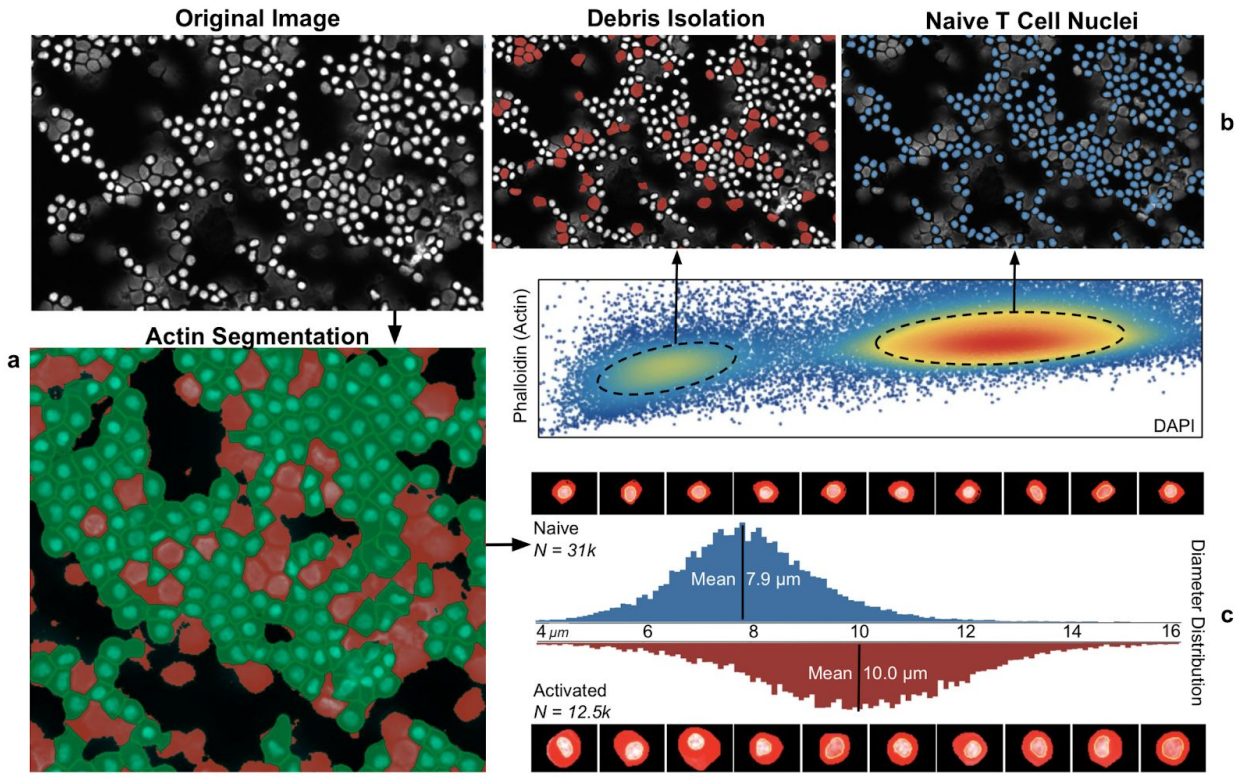

Figure 9: Cell diameter recovery workflow as annotated Explorer screenshots. **(a)** Unstimulated/naive cells with DAPI stain (gray), binary actin image (red), and resulting cell segmentation (green). **(b)** Projection of modes in Phalloidin/DAPI distribution to segmented cells in original image. **(c)** Diameter distribution comparison for unstimulated and activated samples along with corresponding single cell images.

Much like the previous section demonstrating estimation of CD4+/CD8+ T cell population sizes, obtaining accurate cell size distributions was found to be difficult without first establishing proper filters to eliminate debris, lysed cells, and poorly segmented nuclei. Similar procedures for assays based on dissociated cells, e.g. Flow and Mass Cytometry, are far more formalized yet satisfactory results can still be obtained with image cytometry as long as the relationships between spatial and expression dimensions can be properly characterized. **Figure 10** demonstrates results from a workflow built to isolate singlet cells in Explorer and then applied to a variety of human and mouse cell cultures at differing levels of magnification. Inferred cell size distributions are shown alongside single estimates of average cell diameters from a dedicated device.

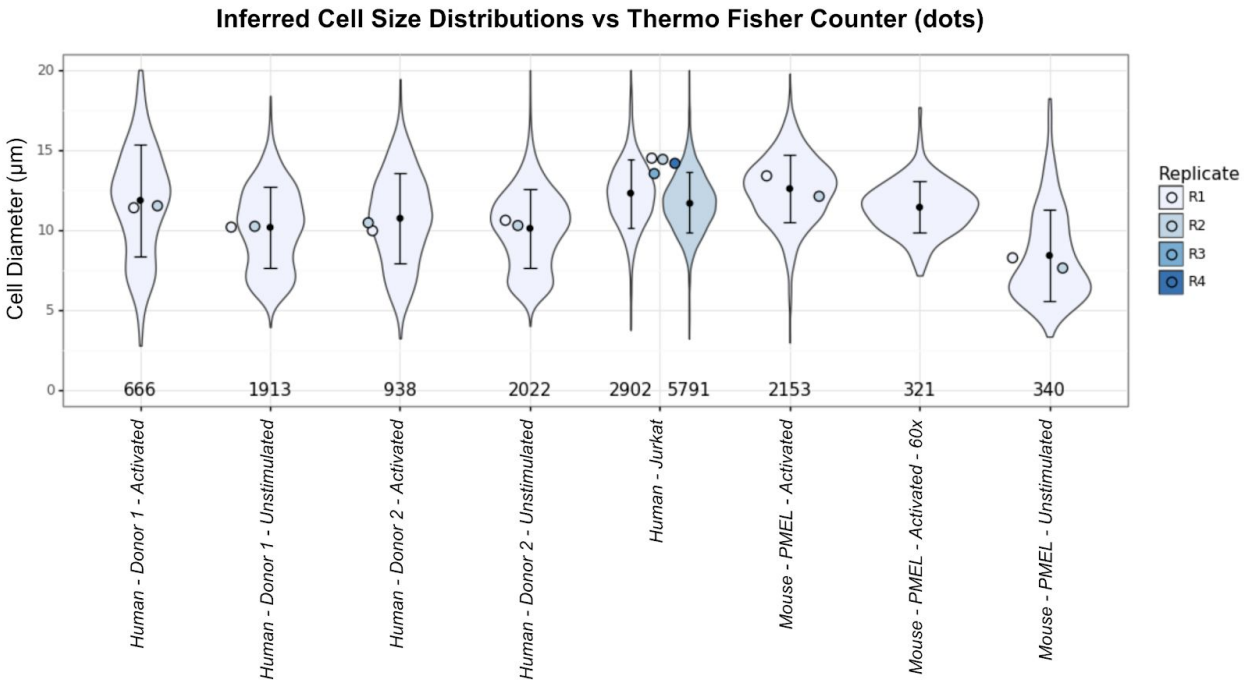

Figure 10: Inferred cell diameter distributions (violin) vs point estimates from [Thermo Fisher Countless II Automatic Cell Counter] (dot) as well as filtered single cell population sizes (counts); All images taken at 20x magnification except where indicated otherwise.

CODEX

Representative preparations of image datasets resulting from CODEX protocol applications were also carried out to characterize performance at a larger scale. The first of these included an analysis of the [reference dataset] shared by the CODEX authors which, in its totality, contains ~700k cells spread across 9 individual tissue preparations and ~1.1TB of microscope images. These images, of normal and autoimmune murine spleens, were prepared as separate replicates of tissue slides with a single replicate being subjected to 18 cycles of imaging (on a 9x7 grid) to capture 54 expression signals. Cytokit was applied to a single replicate containing 51,030 images (129GB) to demonstrate that cycle alignment, deconvolution, image quality assessment, and segmentation can be executed in ~80 minutes on one (2x Nvidia GTX 1080 GPU) workstation. Segmentation results*[1] were visually and quantifiably similar to those in the original study (**Figure 11**).

[1] CODEX reference data was already aligned and deconvolved so segmentation was applied separately from preprocessing operations used for time benchmarks

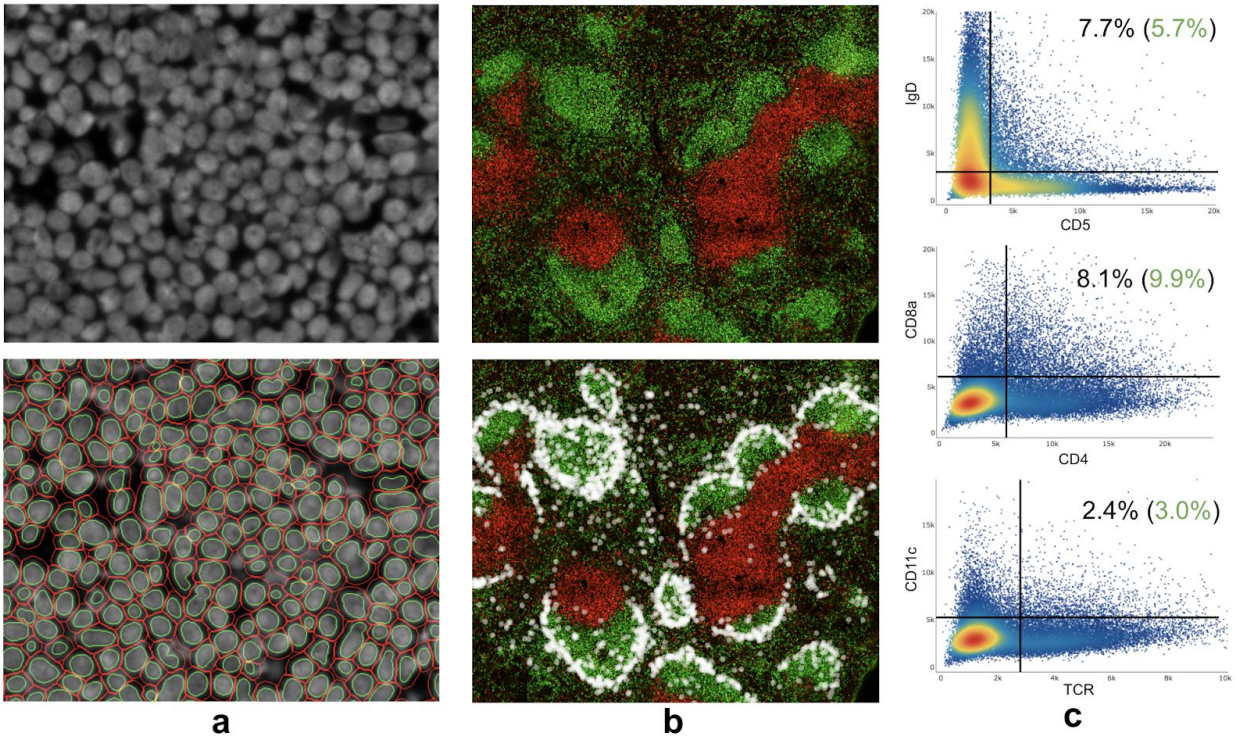

Figure 11: BALB/c spleen (slide 1) CODEX image segmentation/quantification results. **(a)** ~200 cells with DRAQ5 nuclear stain (above) and corresponding cell/nucleus segmentation (below). **(b)** ~71k cells in stitched, downscaled 9072x9408 pixel image showing IgD (green) and CD90 (red) expression as well as location of CD169+ marginal zone macrophages as white dots. **(c)** Double positive cell population rates post cleanup-gating with expected percentages in green

Conclusion

Multiplexed, in-situ image cytometry offers many potential advantages over traditional single cell assays, but the volume of image data produced by these procedures makes processing and analyzing them difficult. Cytokit is an open source Python package that attempts to help overcome these issues by providing GPU-accelerated implementations of cycle registration and image deconvolution as well as an interactive user interface for characterizing the phenotypic features of cells alongside their spatial distribution. We find that understanding modalities in expression distributions as projections onto original images and the ability to iteratively optimize pipelines operating on large datasets are both important capabilities needed to achieve good results, even for the relatively simple tasks presented here such as size estimation and cellular marker profiling.

Acknowledgements

Supported in part by the Flow Cytometry and Cell Sorting Unit Shared Resource, Hollings Cancer Center, Medical University of South Carolina (P30 CA138313).

We would also like to thank River Abedon for his contributions to our early image segmentation and modeling efforts.

Supplemental Materials and Methods

Human primary T cells were isolated, cultured, activated, and prepared for immunofluorescence microscopy as previously described [21]. In short: for sample preparation for imaging, T cells that were isolated from buffy coats obtained from healthy donors, cyto-spun onto sample cover slips, processed according to suggested CODEX sample preparation protocols (Goltsev et al. 2018), and then mounted onto a glass slide with the Prolong anti-fade mounting reagent. For samples stained with only DAPI and phalloidin, T cells from cultures were washed and re-suspended in PBS, fixed in 4% formaldehyde solution for 30 minutes, washed and stained with the phalloidin dye for 30 minutes within BD CytoPerm solution, and re-suspended in PBS before cytospinning. Human primary T cells were activated with anti-CD3 and anti-CD28 beads for two days and de-beaded before being prepared for the imaging. T cells from pmel-1 mouse were activated by adding 1 uM of gp100 peptide to freshly isolated splenocytes and culturing for three days.

References

1. Spitzer MH, Nolan GP. Mass Cytometry: Single Cells, Many Features. Cell. 2016;165: 780–791.

2. Feher K, von Volkmann K, Kirsch J, Radbruch A, Popien J, Kaiser T. Multispectral flow cytometry: The consequences of increased light collection. Cytometry A. 2016;89: 681–689.

3. Blom S, Paavolainen L, Bychkov D, Turkki R, Mäki-Teeri P, Hemmes A, et al. Systems pathology by multiplexed immunohistochemistry and whole-slide digital image analysis. Sci Rep. 2017;7: 15580.

4. Angelo M, Bendall SC, Finck R, Hale MB, Hitzman C, Borowsky AD, et al. Multiplexed ion beam imaging of human breast tumors. Nat Med. 2014;20: 436–442.

5. Goltsev Y, Samusik N, Kennedy-Darling J, Bhate S, Hale M, Vazquez G, et al. Deep Profiling of Mouse Splenic Architecture with CODEX Multiplexed Imaging. Cell. 2018;174:

968–981.e15.

6. Wang Y, Woehrstein JB, Donoghue N, Dai M, Avendaño MS, Schackmann RCJ, et al. Rapid Sequential in Situ Multiplexing with DNA Exchange Imaging in Neuronal Cells and Tissues. Nano Lett. 2017;17: 6131–6139.

7. Lin J-R, Izar B, Wang S, Yapp C, Mei S, Shah PM, et al. Highly multiplexed immunofluorescence imaging of human tissues and tumors using t-CyCIF and conventional optical microscopes. Elife. 2018;7. doi:10.7554/eLife.31657

8. Hedley BD, Keeney M. Technical issues: flow cytometry and rare event analysis. Int J Lab Hematol. 2013;35: 344–350.

9. 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. 2017;18: 529.

10. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems [Internet]. 2015. Available: https://www.tensorflow.org/

11. Guizar-Sicairos M, Thurman ST, Fienup JR. Efficient subpixel image registration algorithms. Opt Lett. 2008;33: 156–158.

12. van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. scikit-image: image processing in Python. PeerJ. 2014;2: e453.

13. Yang SJ, Berndl M, Michael Ando D, Barch M, Narayanaswamy A, Christiansen E, et al. Assessing microscope image focus quality with deep learning. BMC Bioinformatics. 2018;19: 77.

14. Sage D, Donati L, Soulez F, Fortun D, Schmit G, Seitz A, et al. DeconvolutionLab2: An open-source software for deconvolution microscopy. Methods. 2017;115: 28–41.

15. Li J, Xue F, Blu T. Fast and accurate three-dimensional point spread function computation for fluorescence microscopy. J Opt Soc Am A Opt Image Sci Vis. 2017;34: 1029–1034.

16. McQuin C, Goodman A, Chernyshev V, Kamentsky L, Cimini BA, Karhohs KW, et al. CellProfiler 3.0: Next-generation image processing for biology. PLoS Biol. 2018;16: e2005970.

17. Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. 2006;7: R100.

18. Eric Jones and Travis Oliphant and Pearu Peterson and others. SciPy: Open source scientific tools for Python [Internet]. 2001. Available: http://www.scipy.org/

19. Bradski G. The OpenCV Library. Dr Dobb's Journal of Software Tools. 2000;

20. Finak G, Frelinger J, Jiang W, Newell EW, Ramey J, Davis MM, et al. OpenCyto: an open source infrastructure for scalable, robust, reproducible, and automated, end-to-end flow cytometry data analysis. PLoS Comput Biol. 2014;10: e1003806.

21. Aksoy BA, Aksoy P, Wyatt M, Paulos C, Hammerbacher J. Human primary T cells: A practical guide [Internet]. 2018. doi:10.7287/peerj.preprints.26993v1