Quantifying Variability in Microscopy Image Analyses for COVID-19 Drug Discovery

Mylene Simon
National Institute of Standards and Technology
100 Bureau Drive Gaithersburg, MD 20899
Mylene.Simon@nist.gov

Nicholas J. Schaub
National Institutes of Health (NIH) - National Center for Advancing Translational Science
9800 Medical Center Dr, Rockville, MD 20850
Nick.Schaub@nih.gov

Sunny Yu
National Institutes of Health (NIH) - National Center for Advancing Translational Science
9800 Medical Center Dr, Rockville, MD 20850
Sunny.Yu@nih.gov

Mohamed Ouladi
National Institutes of Health (NIH) - National Center for Advancing Translational Science
9800 Medical Center Dr, Rockville, MD 20850
Mohamed.Ouladi@nih.gov

Jayapriya Nagarajan
National Institutes of Health (NIH) - National Center for Advancing Translational Science
9800 Medical Center Dr, Rockville, MD 20850
Jayapriya.Nagarajan@nih.gov

Sudharsan Prativadi Bayankaram
National Institutes of Health (NIH) - National Center for Advancing Translational Science
9800 Medical Center Dr, Rockville, MD 20850
Sudharsan.Prativadi@axleinfo.com

Peter Bajcsy
National Institute of Standards and Technology
100 Bureau Drive Gaithersburg, MD 20899
Peter.Bajcsy@nist.gov

Nathan Hotaling
National Institutes of Health (NIH) - National Center for Advancing Translational Science
9800 Medical Center Dr, Rockville, MD 20850
Nathan.Hotaling@nih.gov

2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) | 978-1-6654-4899-4 /21/$31.00 USG | DOI: 10.1109/CVPRW53098.2021.00421

Abstract

Microscopy image-based measurement variability in high-throughput imaging experiments for biological drug discoveries, such as COVID-19 therapies was addressed in this study. Variability of measurements came from (1) computational approaches (methods), (2) implementations of methods, (3) parameter settings, (4) chaining methods into workflows, and (5) stabilities of floating-point arithmetic on diverse hardware. Measurement variability was addressed by (a) introducing interoperability between algorithms, (b) enforcing automated capture of computational provenance and parameter settings, and (c) quantifying multiple sources of variabilities for 10 nucleus measurements, from 8 workflow streams, executed in 2 workflow graph configurations, on 2 computational hardware platforms at 2 locations. Using modified Mean Absolute Error (mMAE [%]) to compare measurements, We concluded that for the task of image-based nucleus measurements the variability sources were (1) implementations (0.10 % - 5.72 % per measurement), (2) methods (3.08 % - 3.11 % between Otsu thresholding and CellPose segmentation), (3) parameters (1.16 %-1.17 % between 4- and 8-neighbor connectivity), (4) workflow graph construction and computer hardware (negligible).

1. Introduction

Variability problems in image-based measurements have been documented in reports of irreproducibility in the Artificial Intelligence/Machine learning (AI/ML) field. Most recently, two new analyses [1], [2] put the spotlight on machine learning in health research, where lack of reproducibility and poor quality could risk harm to patients and/or lower the quality of care a patient receives. For example, Roberts et al. in [2] showed that out of the 415 models for classifying X-Rays and CTs of patients with COVID-19 that they tried to reproduce, only 62 passed two standard reproducibility and quality checklists, CLAIM [3] and RQS [4]. Of the remaining 62, including two currently in use in clinics, the team found that none were developed without significant biases in study design and methodological flaws. Further, McDermott et al.in [1] showed that only 20 % of machine learning in healthcare (MLH) papers made their code available, 55 % made their dataset available, 40 % provided model variance in performance, and 23 % of papers used multiple datasets to confirm their results. Both papers [1], [2] highlight that platforms and technologies that can help overcome these shortcomings are critical to the field.

Our motivation was to improve reproducibility of image-based measurements and quantify the measurement

978-1-6654-4899-4/21/$31.00 ©2021 IEEE
DOI 10.1109/CVPRW53098.2021.00421

uncertainty while leveraging all cutting-edge approaches to image-based drug discoveries. Here, we selected measurements derived from fluorescently labeled nucleus images over samples with a variety of drug treatments for COVID-19. Such measurements require (1) nucleus segmentation from background, (2) labeling each nucleus with a unique label, and (3) computing intensity and shape measurements per unique label of a nucleus. Of the sources of irreproducibility discussed in [1] and [2] poor study design cannot be overcome with software alone. However, sources of irreproducibility that could be measured and mitigated using software include (a) interoperability of cutting-edge approaches and their implementations, (b) automatic capturing of computational provenance about input and intermediate datasets, as well as parameter settings, (c) understanding of variabilities due to a spectrum of computational workflows, and (d) sharing resources to compare reproducibility across many hardware and software workflow solutions due to the large volume of data and time-consuming complex computations.

2. Methods

We approached the variability quantification problem by (a) introducing interoperability of containerized software methods [5] across the teams at the National Institute of Standards and Technology (NIST) and the National Center for Advancing Translational Science – National Institutes of Health (NCATS), (b) collaborating on client-server platforms called WIPP [6] and Polus [7] that can automatically capture computational provenance while executing workflows, (c) estimating the variabilities of nucleus measurements by comparing results across multiple workflow streams on small data subsets for the same task, and (d) sharing computer cluster and cloud resources for quantifying measurement variabilities over large image datasets.

Our experiments included running 8 workflow streams (WS) consisting of (1) two segmentation methods, (2) three labeling methods, (3) two feature extraction libraries, and (4) two workflow graphs executed on two computational platforms. Each WS is a chain of segmentation, labeling and feature extraction steps drawn from the seven methods summarized in Table 1. Three methods were implemented by the NIST team and four methods were implemented by the NCATS team while following the guidelines for interoperability of containerized software [5]. The WSs were combined into one workflow at NCATS as shown in

Table 1: The list of implemented tools used for chaining as steps of constructed WSs for computing 2D image-based nucleus measurements

Segmentation	Labeling	Feature Extraction
M1: Otsu Thresholding[8] Version: Thresholding plugin:1.1.1	L1: Raster Java Mask Labeling[9] Version: WIPP Mask labeling plugin:0.0.2 (connectivity = 4 nbh) L2: Raster Faster Than Light Labeling[12] Version: FTL Label:0.2.2 (connectivity = 1 or 2) L3: Vector label[11] Vector-label-plugin:0.2.5 (flowThreshold = 0.8, cellProbThresh=0, stitchThresh=0)	F1: Java-based Feature Extraction[10] Version: WIPP Feature2DJava Plugin:1.5.0
M2: CellPose Segmentation[11] Version: Cellpose-inference :0.3.4 (diameter=22,nuclei pretrained model)		F2: Python-based (Sci-kit Image) Feature extraction[13] Version: Feature Extraction:0.10.0

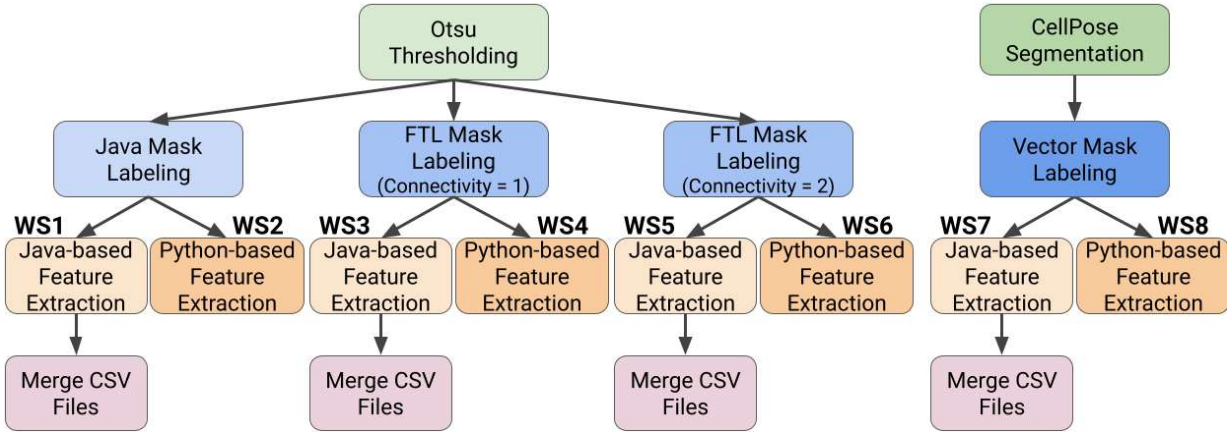

Figure 1: Summary of executed workflow generating 2D measurements per nucleus including mean intensity, area, perimeter, centroid, bounding box information, and gray level co-occurrence matrix (GLCM)-based texture entropy (average)[14]. The steps are denoted according to Table 1.

Figure 1 or kept separate at NIST, and then executed on NIST and NCATS computational resources independently.

The nucleus measurements include mean intensity, area, perimeter, centroid, bounding box information, and intensity entropy [14]. All measurements were evaluated by using the modified mean absolute error (mMAE) in percent as defined in Equations 1 and 2.

$$\mu_{i,j} = \frac{1}{n}\sum_{k=1}^{n} x_{i,j,k} \quad (1)$$

$$mMAE[\%] = \left| \frac{\mu_{i,j} - \mu_{i+1,j}}{0.5 * \left(\mu_{i,j} + \mu_{i+1,j}\right)} * 100 \right| \quad (2)$$

Where j is the index of a measurement (nucleus feature) and j∈[1,10]; i is the index of a workflow stream and i∈[1,8]; k is the index of a segmented region and k∈[1,n], n is the number of nuclei (segmented regions) per image, x_(i,j,k) is a measurement, and μ_(i,j) is an average of measurements over all regions. Eq (2) for mMAE can be related to the differences normalized by the average as used in [18] (Eq. 7.1),but is executed over averages defined in Eq (1). mMAE was used because there was no one-to-one correspondence between segmentation regions and therefore averages of features per WS were computed for each comparison. When comparing identical WSs between institutions, or when comparing between feature extraction libraries standard mean squared error (MSE) was used because there was one-to-one correspondence of regions.

2.1. Datasets and computational platforms

The 8 WSs in Figure 1 were applied to image collections downloaded from the Recursion company website [15]. We processed the nuclear channel (s1) of the human umbilical vein endothelial cells (HUVEC). The HUVEC-1 and HUVEC-2 cell line subsets of the RxRx2 dataset t contain a total of 131 953 images (1024 x 1024 pixels per image, 8 bits per pixel, PNG file format). The two HUVEC cell lines were treated with 464 immunomodulating compounds at six concentrations spanning: chemokines, checkpoint inhibitors, growth factors, immunoglobulins, cytokines, etc. [16]. The goal of the treatments was to profile cellular response via image analysis to appropriately cluster immune perturbations by function and to rapidly employ these states for high-throughput drug screening applications for relevant therapies of COVID-19.

The WSs were executed at NIST in a WIPP deployment on a single desktop with 128 GiB of RAM, 32 CPU processors (Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz, 20.48MB cache). The same workstreams were launched at NCATS in a Polus deployment on Amazon Web Services (AWS) cloud resources - 4x nodes of the EC2 R5 instances with 8 vCPU and 32 GiB RAM.

Figure 2 shows the user interfaces for constructing the total workflow and the callout shows the hyperlinked provenance enabled by the platform. Each step contains all input data locations, output data locations, and parameter settings to simplify reproducibility of workflows. The measurement results from the workflows were tabular and downloaded in a CSV file format from the web systems deployed at NIST and NCATS. Due to the use of different labeling methods, the same nucleus was assigned different unique labels by each labeling method. Thus, to match results for the same nucleus, the nuclei were matched by sorting the CSV columns by “Workflow", "Centroid_X", and "Centroid_Y" in that order of hierarchical sorting.

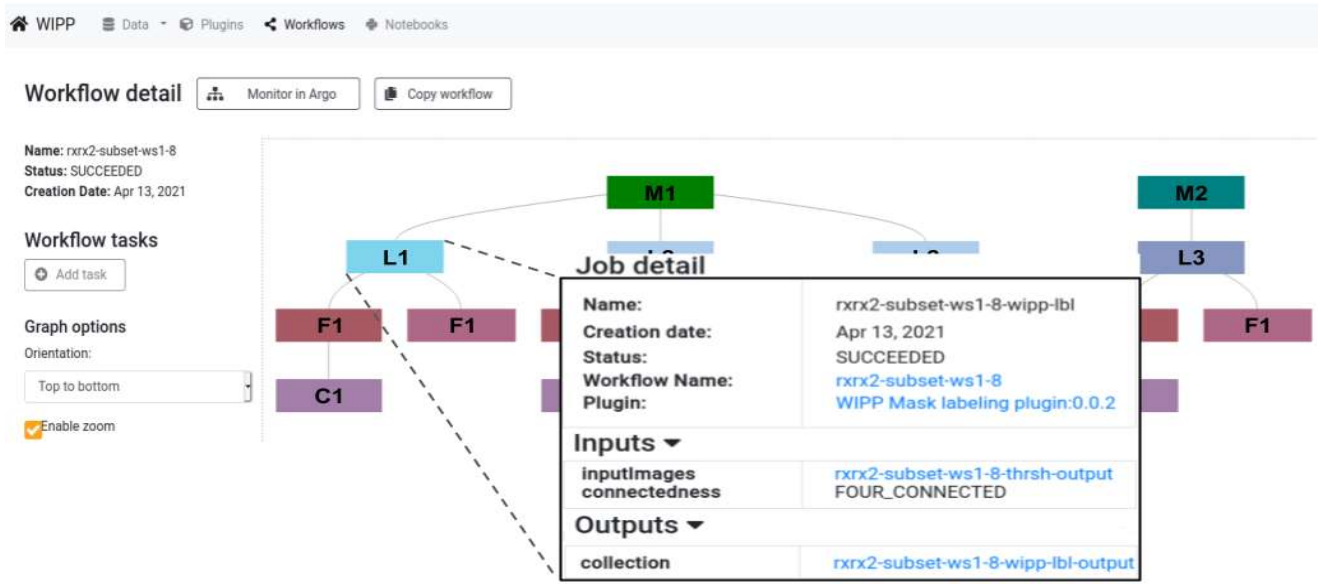

Figure 2: Web-based user interface for constructing one workflow combining the eight workflow streams shown in Figure 1, with callout for hyperlinked provenance and parameter settings at each step in the workflow.

3. Results

3.1 Execution times

The nucleus measurements listed in Figure 1 (caption) were collected on a small dataset (20 images) to estimate (a) the sources of variability and (b) required computational time per image. The 20 images were selected from HUVEC-1 cell line, Row AA, columns 02-07. Table 2 shows an overview of execution times and the total number of found nuclei for each WS at NIST and NCATS. The re-executions did not change the time benchmarks at the reported accuracy. Based on the results in Table 2, we could choose one of the workstreams and estimate the execution time for processing the entire dataset of two HUVEC cell lines.

Table 2: Execution time per image and a total number of detected nuclei in 20 images drawn from the RxRx2 data subset.

Workflow Stream	NIST (s/image)	NCATS (s/image)	Total # of Regions (nuclei)
WS1	1.4	5.1	7620
WS2	3.0	4.8	7620
WS3	1.2	3.2	7620
WS4	1.3	2.9	7620
WS5	1.3	3.1	7456
WS6	1.9	2.8	7456
WS7	21.0	65.6	7487
WS8	21.0	65.3	7487

3.2 Variability of measurements due to workflow construction and computational hardware

We compared the average nucleus measurements from 8 WSs shown in Figure 1 that were computed at NIST and NCATS. Methodological differences between the two were (a) computational hardware specifications and (b) constructed workflows that execute WSs one-by-one at NIST or one main workflow combining all WSs at NCATS (see Figure 2) and (c) different versions of the Argo workflow scheduler [19] (Argo 2.4.2 at NIST and Argo 2.2.1 at NCATS). To validate the variability due to these three sources extracted feature values were compared between NIST and NCATS.

Table 3 shows the minimum and maximum MSEs per measurement between NIST and NCATS. No MSE was greater than 10E-25 and relative to the values of these features in biological experiments the magnitude of MSE is negligible in comparison. It was therefore concluded that the variability from the three sources of variability described above was negligible. The small differences in MSEs were related to the numerical reproducibility of floating-point arithmetic and the stability of implementing average calculations in different implementations [17]. We conclude that the magnitude of MSE is negligible in comparison to other sources of variability in biological experiments, and thus that variability from different hardware platforms executing the same containerized code in WSs is negligible using our containerized approach. Minimizing these variances is seen as a strong recommendation for using these types of systems as it has been reported that variance of results based on diverse hardware can be large [20].

Table 3: Comparisons of extracted nucleus measurements over the 8 WSs executed at NIST and NCATS on different computational platforms by computing mean squared errors (MSEs) per workstream. The MSE units are [pixel^2] or [intensity^2].

Comparison	Min MSE	Max MSE
Bounding Box	0.00	0.00
Area	0.00	0.00
Centroid	0.00	3.57E-25
Average Intensity	0.00	0
Perimeter	0.00	7.15E-27
Entropy	0.00	8.39E-30

3.3 Variability of measurements due to segmentation methods:

The variability of 2D measurements between the Otsu thresholding and CellPose segmentation methods (M1 and M2 in Table 1) was compared next. This maps to workstream groupings of WS1-WS6 versus WS7-WS8 in Figure 1. Figure 3 illustrates the key differences between the measurements from WS1-WS6 and WS7-WS8. We concluded that the CellPose segmentation method (WS7-WS8) has much less variability in all morphological measurements than the unsupervised Otsu thresholding method (WS1-WS6) however intensity and location metrics (Mean and Centroid X/Y in Figure 3) showed no overall differences between WS1-WS6 and WS7-WS8. This is unsurprising as segmentation morphology is heavily influenced by the algorithm chosen for segmentation while the pixel intensities these masks are overlaid on remain constant, regardless of mask, and therefore average values remain relatively unchanged over large pixel collections. Additionally, this observation implies that the supervised CellPose segmentation has learned size limits during its training (area and bounding box width and height) and shape constraints (perimeter) that are applied as filters to the resulting segments.

Quantitative comparisons of the two segmentation methods can also be derived from Figure 4, for example by comparing the average mMAE across all features between WS1 and WS7 (mMAE=3.08 %) or WS2 and WS8 (mMAE= 3.11 %). These results were not surprising as it is well known that different segmentation methods can lead to dramatically different feature values. However, in the context of nuclear segmentation it is important to note that the unsupervised Otsu method did not, on average, perform that differently from the state-of-the-art CellPose algorithm. Therefore, an assessment of the necessary measurement

variance should be done for each application to understand if the computational trade-off (12x to 15x Table 2) is worth it. For biomedical applications where people's lives can be the cost it was concluded that the lower variance, higher computational cost, CellPose was worth the additional computational burden.

3.4 Variability of measurements due to connectivity parameter settings in raster mask labeling methods

This variability (mMAE) was computed between WS3-WS4 (connectivity=1, 4-neighbors) versus WS5-WS6 (connectivity=2, 8-neighbors). Figure 4 shows that the mMAE values for WS3-WS5 and WS4-WS6 were 1.17 % and 1.16 % respectively. These workflows compare the variability due to connectivity in the same Faster Than Light (FTL) labeling method and thus are the most conservative. The variance in "connectedness" of labels is dependent on the density of objects that are being segmented. In the images chosen for validation, nuclear density was low and therefore, 4- or 8-connected labeling did not have a large

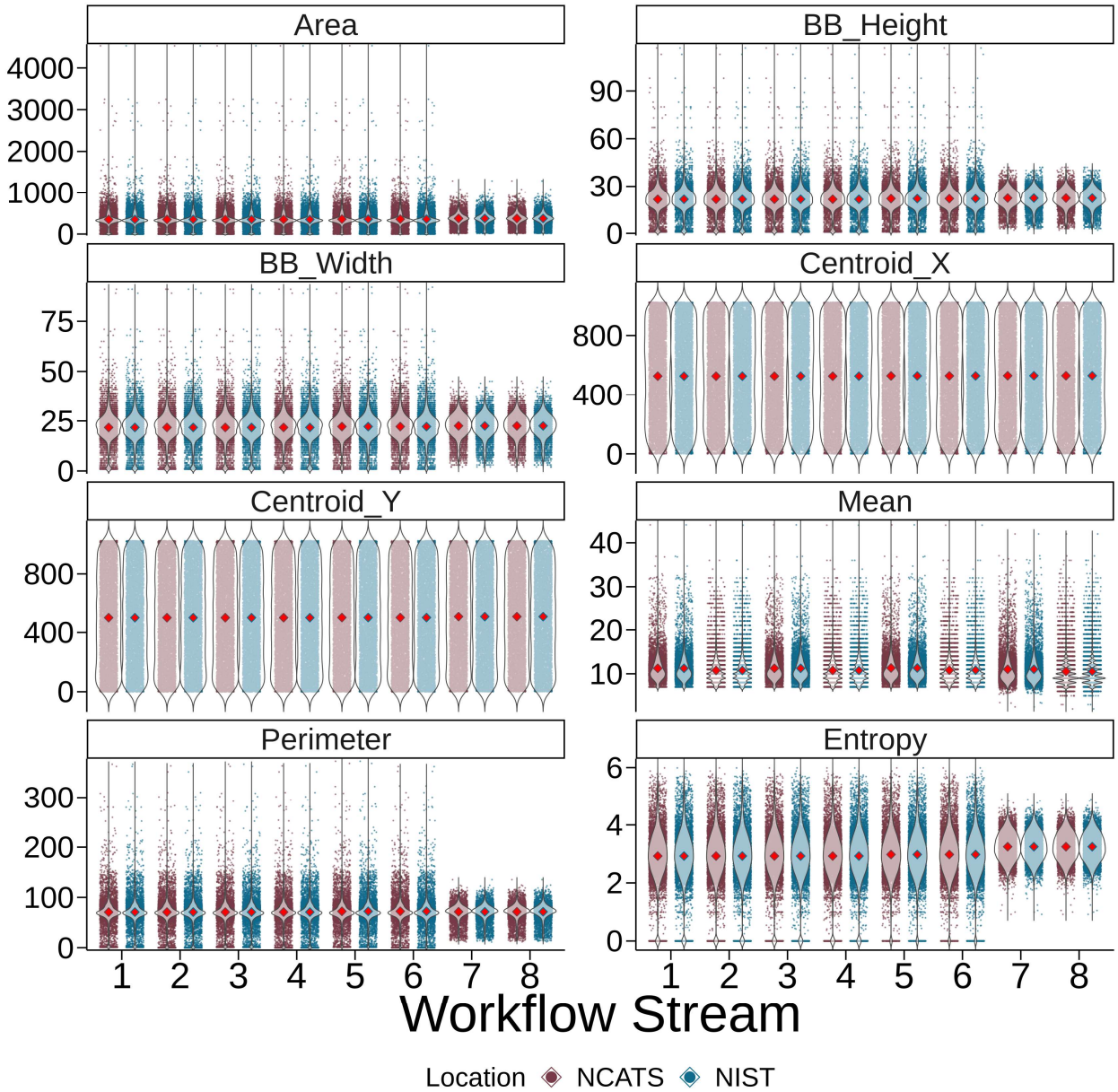

Figure 3: Comparison of 8 measurements of nuclei derived from 8 workstreams at NIST and NCATS. Each blue/maroon dot is a segmented region and the bright red diamonds correspond to the average of all values. Area measurements were in pixels squared, BB_Height, BB_Width, Centroid_X, Centroid_Y, and Perimeter were all in pixels, Mean Intensity and Entropy were both in pixel intensity units.

impact. However, it can be seen in Table 2 that with 8-connected assessment a decrease in the number of distinct regions occurred. These regions were ostensibly close together and merged together when the more stringent connected method was applied. This type of occurrence would happen much more frequently in dense cell culture conditions, where nuclei would be much closer together, and therefore the differences between label connection would be much higher. Due to the inherent variability of cell culture, and the desire of the measurement to be as discerning as possible, it is recommended to use the less stringent, 4-connected method as it differentiates between close objects with a higher fidelity than the 8-connected method. **Note**: the WS1-WS2 workstreams using the Java Mask Labeling method have the connectivity setting equivalent to WS3-WS4 (4-connected) using the FTL labeling method. Thus, WS1 and WS3, as well as WS2 and WS4, yield identical results, Figure 4.

3.5 Variability of measurements due to feature extraction implementations

This variability is computed between pairs of WSs: WS1-WS2, WS3-WS4, WS5-WS6, and WS7-WS8 (Java-based Feature extraction versus Python-based Feature Extraction) as shown in Figure 1. Differences across metrics can be seen in Table 4 and Figure 5. For the 4 pairwise WS comparisons, area, bounding box width and height, and bounding box location values were found to be identical and thus were not included in Table 4. Additionally, Centroid X and Centroid Y coordinates as well as Intensity Entropy were found to only vary by floating point precision and therefore, while recorded in Table 4 were not deemed significant. Interestingly, Mean Intensity as well as Perimeter were both found to have significant deviance between the reported workstreams.

Mean Intensity was found to vary due to how each library computes their respective values. The Python implementation was derived from the Sci-Kit Image library which outputs an identical data type as that which was used as input. Therefore, since the input images were 8 bits per pixel represented as integer values, the output was an integer value rather than a float64 data type. The Java based library did all calculations as a float64 data type. Thus, the discrepancy between these two libraries for Mean Intensity highlights how measurement variance can be introduced even if the algorithm to calculate a value is identical between two libraries. Data variable typing in algorithms is not frequently well understood by the biological community and thus measurement variance can be introduced without a biologist/clinician being aware of the trade-off in accuracy that is being made. The authors recommend all calculations be performed as float64 and have subsequently modified the implementation of the Python library to eliminate the numerical differences following this recommendation.

Perimeter was also found to vary between workflows. According to [14], this variability can come from an

Table 4: Comparisons of feature extraction implementations. All values are in MSE +/- standard deviation.

Feature	WS1-WS2	WS3-WS4	WS5-WS6	WS7-WS8	Avg.
Centroid_X	1.72E-25 ± 1.43E-24	1.72E-25 ± 1.43E-24	1.79E-25 ± 1.47E-24	1.74E-25 ± 1.41E-24	1.74E-25 ± 1.44E-24
Centroid_Y	1.52E-25 ± 1.31E-24	1.52E-25 ± 1.31E-24	1.54E-25 ± 1.32E-24	1.5E-25 ± 1.28E-24	1.52E-25 ± 1.3E-24
Intensity Entropy	4.38E-30 ± 7.53E-30	4.38E-30 ± 7.53E-30	4.46E-30 ± 7.58E-30	4.41E-30 ± 7.09E-30	4.41E-30 ± 7.43E-30
Mean Intensity	0.319 ± 0.30	0.319 ± 0.30	0.325 ± 0.30	0.329 ± 0.294	0.323 ± 0.298
Perimeter	0.604 ± 4.58	0.604 ± 4.58	0.724 ± 7.62	0.0755 ± 0.397	0.502 ± 4.29

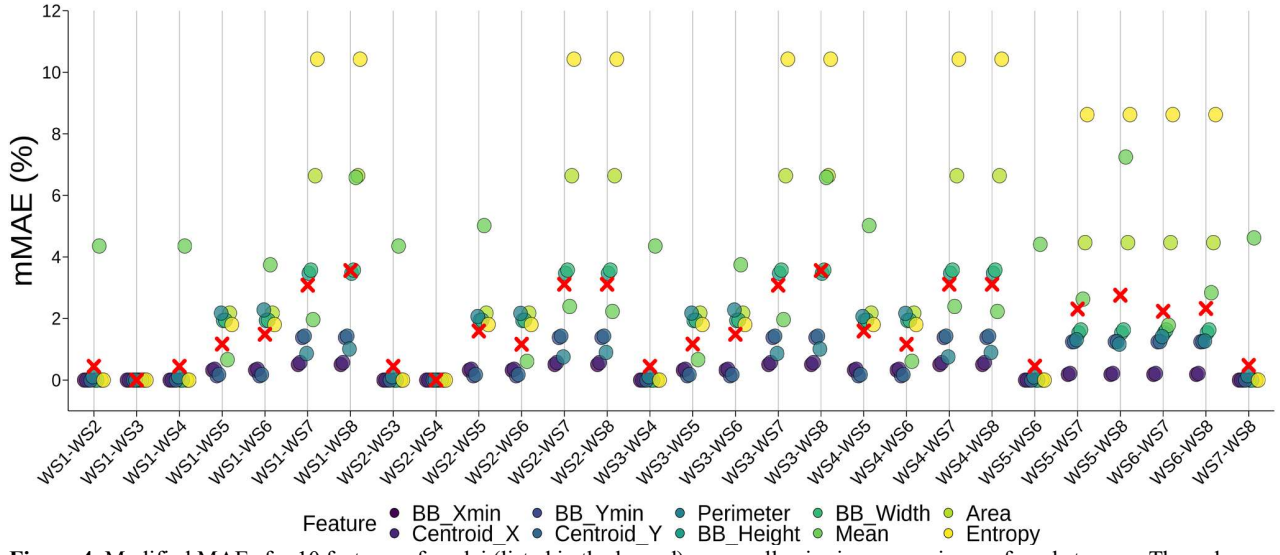

Figure 4: Modified MAEs for 10 features of nuclei (listed in the legend) across all pairwise comparisons of workstreams. The red cross corresponds to the average.

ambiguous definition of perimeter and the discretization of pixels (e.g., a perimeter estimated from inside or outside of a pixel or using Manhattan vs Euclidean length). We hypothesize that the non-zero MSE values in our study originate from Java and Python image representations and feature implementations being different in these basic assumptions. To identify the exact sources is the topic of our future work.

When analyzing 4 pairwise comparisons differing only in the feature extraction libraries used, the MSE remains fairly consistent (Table 4 across any row). Thus, the authors determined that feature extraction library variance was relatively insensitive to variations in connectedness, segmentation method, or feature labeling algorithm. A notable exception was for perimeter where the CellPose segmentation (WS7-WS8) had an order of magnitude lower MSE than the Otsu thresholding comparisons. This difference can be accounted for because Perimeter is sensitive to the outer set of pixels for a given region. The smaller the region, the more the pixel discretization error increases [14] as greater and greater percentages of the border no longer follow a smooth gradient that can be adequately captured via discretized pixel units. Therefore, segmentation methods that limit the number of smaller regions will yield lower measurement variance. Cellpose was seen to dramatically reduce small regions, as shown in Figure 3, and thus its lower variance in Table 4 is consistent with expected trends in the perimeter algorithm.

3.6 Measurements for Drug Discovery from RxRx2 datasets

We executed WS5 as one the fastest workflow streams on the entire large RxRx2 nuclear channel dataset (LWS5) and completed the calculations in 14 h at NIST and 12.5 h at NCATS. For LWS5, the variabilities due to (a) Otsu thresholding method vs CellPose segmentation (WS5-WS7) was 2.31 % mMAE, (b) connectivity parameter set to 4- vs 8-neighbors in the Faster Than Light (FTL) labeling method (WS3-WS5) was 1.17 % mMAE, and (c) Java-based vs Python-based feature extraction method (WS5-WS6) was between 0.10 % and 5.72 % mMAE per measurement which agreed well with results shown in Figures 4 and 5. As we are applying clustering methods to profile cellular response to immune perturbations by function, we are able to assign these variabilities to the obtained clusters and propagate them into our confidence of drug discovery relevant interpretations. The statistical significance of clustering-based conclusions is captured by analyzing over 498 million unique measurements (10 measurements per nucleus) in a single workstream from the RxRx2 dataset.

4. Discussion

The goals of this report were to summarize, categorize, understand, and recommend best practices for the analysis of high content drug screening. To this end, we found that when using our container-based platforms, adhering to interoperable container plugins standards, and sharing computational processing algorithms and provenance information, we were able to mitigate measurement variance due to hardware differences, workflow stream execution order, and workflow scheduler version (Table 2).

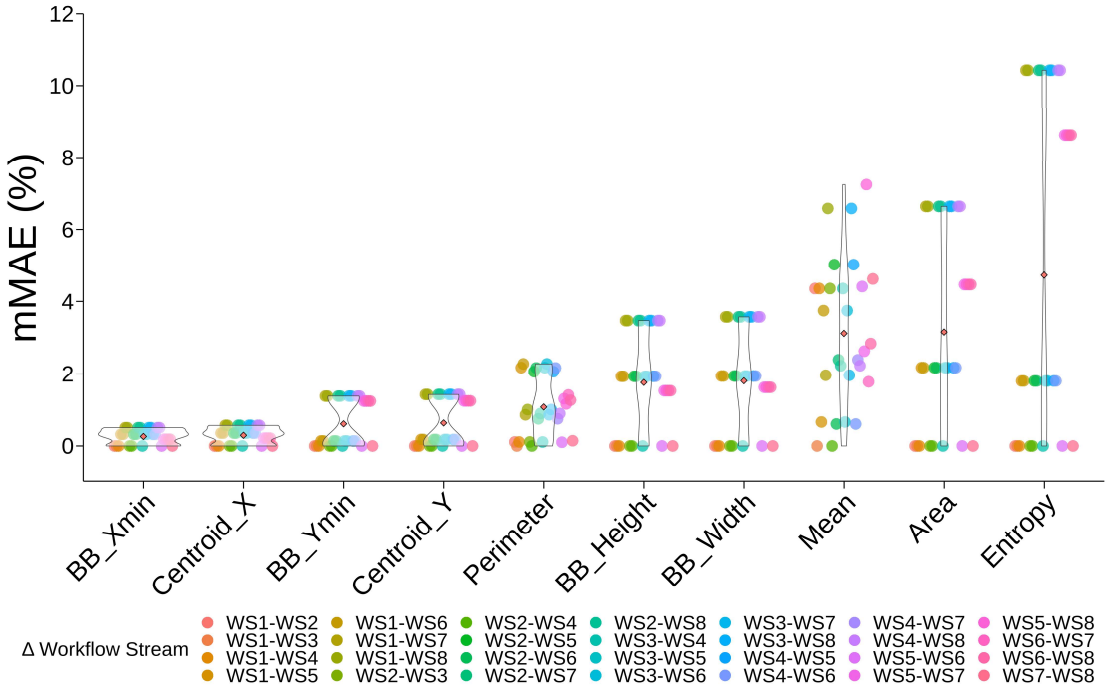

Figure 5: Measurement variability across all pairwise workflow stream comparisons.

Of note, is that eliminating variance due to hardware and software platforms is an important step in reproducibility [20].

Additionally, we quantified the variability that can occur due to various segmentation methodologies. Here we unsurprisingly found that segmentation methods can dramatically alter the variance of segmentation regions (Figure 3). However, we also found that in the specific application of nuclear segmentation, mean values did not shift dramatically between methods. Thus, an assessment of the sensitivity of the assay desired in the application needs to be performed in order to justify the added computational cost of more advanced segmentation algorithms, such as the CellPose algorithm implemented in this study. Here, because the application was therapies for a virus that has killed millions of people, the authors find the trade-off in variance worth the additional computational cost. However, not all applications require this level of rigor and it is up to researchers to understand their applications and acceptable levels of variance.

It was also found that the variance in connectivity used by the labeling function can significantly alter measured outcomes (Table 1 and Figure 4). In these low cell density images, the different methods only caused a change of 2-4 % (Figure 4). Nonetheless, in more densely cultured conditions, which are common even within this drug screening assay, differences can grow starker. Thus, the authors recommend using the less stringent, 4-connected method of labeling.

Feature extraction libraries were found to produce measurement variance due to differences in algorithmic implementation (Perimeter) and in integer precision (Intensity Mean). These differences were highlighted most severely in Perimeter for small objects and in objects with low fluorescent intensity, where a decimal rounding due to low precision integer usage leads to a large proportion of the total signal and thus a high error. Therefore, the authors recommend using high precision implementations of algorithms to better address accuracy at all levels of intensity. It is also recommended for assay designers to use magnifications in their assays that allow for salient regions of importance (like the nucleus of a cell) to be relatively large (100s of pixels in size) to minimize errors due to differences in morphological algorithmic implementation.

Finally, using platforms that are designed to enable traceability and reproducibility of computational graphs executed against data was found to dramatically increase the ease of the two institutes replicating each other's experiments in a fast and transparent fashion. With the sharing of a link, we were able to exchange all relevant experimental parameters, execution times, and error logs when bugs were found. Additionally, by using an open-source platforms and algorithms, we were able to identify, understand and correct (in the case of the mean intensity) sources of irreproducibility between workflow streams. These software technologies and methodologies for variability quantification, if enabled for all researchers, would address most of the concerns highlighted in the introduction of this paper other than poor experimental design.

5. Conclusions

The concepts of (a) designing interoperable containerized software tools, (b) sharing workflows with automatically captured parameters and provenance information, and (c) including required compute capabilities are the key aspects in achieving reproducible results across institutions and across a variety of computational platforms. By having software platforms for executing interoperable containerized plugins, nucleus measurements could be not only compared for reproducibility, but also used for uncovering errors (e.g., the mean intensity error or inconsistent notations for row and column coordinates) and any discrepancies in terms of parameter settings (e.g., connectedness or feature names).

The novelty of this work is in quantifying the variability of nucleus measurements from fluorescent images. The variability estimates can be used for improving reproducibility of experimental results across institutions, approaches (methods), algorithmic parameters, and measurement implementations (definitions).The work also presents a framework for assigning variabilities to derived conclusions from image-based measurements in order to increase one's confidence in discovering reliable treatments..

Disclaimer

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

6. Acknowledgements

We would like to acknowledge the teams at NIST and NCATS, as well as all collaborators contributing to the development of interoperable plugins. This research was supported in part by the Intramural/Extramural research program of the NCATS, NIH. The contributing team members include Antoine Gerardin, Michael Majurski, Joe Chalfoun, Philippe Dessau, Walid Keyrouz, Gauhar Bains, Kevin Duerr, Swazoo Claybon, Aaron Friedman, Brett Layman, Hythem Sidky, Mahdi Maghrebi, Melanie Parham, Madhuri Vihani, and Nikita Lysov.

7. References

[1] M. B. A. McDermott, S. Wang, N. Marinsek, R. Ranganath, L. Foschini, and M. Ghassemi, "Reproducibility in machine learning for health research: Still a ways to go," Sci. Transl. Med., vol. 13, no. 586, Mar. 2021, doi: 10.1126/scitranslmed.abb1655.

[2] M. Roberts et al., "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans," Nat. Mach. Intell., vol. 3, no. 3, Art. no. 3, Mar. 2021, doi: 10.1038/s42256-021-00307-0.

[3] J. Mongan, L. Moy, and C. E. Kahn, "Checklist for Artificial Intelligence in Medical Imaging (CLAIM): A Guide for Authors and Reviewers," Radiol. Artif. Intell., vol. 2, no. 2, p. e200029, Mar. 2020, doi: 10.1148/ryai.2020200029.

[4] P. Lambin et al., "Radiomics: the bridge between medical imaging and personalized medicine," Nat. Rev. Clin. Oncol., vol. 14, no. 12, Art. no. 12, Dec. 2017, doi: 10.1038/nrclinonc.2017.141.

[5] P. Bajcsy and N. Hotaling_ "Interoperability of Web Computational Plugins for Large Microscopy Image Analyses," NIST Interagency/Internal Report (NISTIR) – 8297, Mar. 2020. [Online]. Available: https://www.nist.gov/publications/interoperability-web-computational-plugins-large-microscopy-image-analyses.

[6] M. Simon, and P. Bajcsy "Web Image Processing Pipeline (WIPP)", software, NIST, 2021, Accessed: Apr. 08, 2021. [Online]. Available: https://isg.NIST.gov/deepzoomweb/software/wipp

[7] N. Hotaling, N. Schaub, and S. Yu, "Polus", software, NCATS, 2021, Accessed: Apr. 08, 2021. [Online]. Available:https://github.com/LabShare/polus-plugins

[8] P. Bajcsy, J. Chalfoun, and M. Simon, "Web Microanalysis of Big Image Data", Springer International Publishing, 2018.

[9] M. Ouladi and M. Majurski, "WIPP Mask Labeling plugin", software, USNISTGOV GitHub: NIST 2021, Accessed: Apr. 08, 2021. [Online]. Available: https://github.com/usNISTgov/WIPP-mask-labeling-plugin

[10] M. Simon et al., "WIPP plugins", NIST, 2021, Accessed: Apr. 08, 2021. [Online]. Available: https://github.com/usnistgov/WIPP/tree/master/plugins

[11] C. Stringer, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," bioRxiv, p. 2020.02.02.931238, Feb. 2020, doi: 10.1101/2020.02.02.931238.

[12] L. Lacassagne and B. Zavidovique, "Light Speed Labeling for RISC Architectures," in 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 2009, pp. 3245–3248, doi: 10.1109/ICIP.2009.5414352.

[13] S. van der Walt et al., "Scikit-image: Image processing in Python.," PeerJ, vol. 2, no. e453, 2014, [Online]. Available: https://doi.org/10.7717/peerj.453.

[14] NIST, "Gray Level Co-occurrence Matrix (GLCM)-based Texture Entropy Definition." web page, Accessed: Apr. 08, 2021. [Online]. Available: https://isg.nist.gov/deepzoomweb/stemcellfeatures#entropy

[15] Recursion, Inc., "Cellular response and signaling within the immune microenvironment." Recursion download web pages, Accessed: Apr. 08, 2021. [Online]. Available: https://www.rxrx.ai/rxrx2.

[16] M. F. Cuccarese et al., "Functional immune mapping with deep-learning enabled phenomics applied to immunomodulatory and COVID-19 drug discovery," bioRxiv, p. 2020.08.02.233064, Aug. 2020, doi: 10.1101/2020.08.02.233064.

[17] W. Kahan, "Further remarks on reducing truncation errors," Commun. ACM, vol. 8, no. 1, p. 40, Feb. 2018, doi: 10.1145/363707.363723.

[18] P. Bajcsy, J. Chalfoun, M. Simon, M. Kociolek, and M. Brady, "Chapter 7: Object Measurements from 2D Microscopy Images," in Modern Computer Vision for Microscopy Image Analysis, 1st ed., vol. 1, Elsevier Academic Press, 2020.

[19] Apache Project, "Argo Workflows", software, GitHub repository, Accessed: Apr. 08, 2021. [Online]. Available: https://github.com/argoproj/argo-workflows/releases

[20] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," Nat. Biotechnol., vol. 35, no. 4, Art. no. 4, Apr. 2017, doi: 10.1038/nbt.3820.