Cloud-enabled Fiji for reproducible, integrated and modular image processing

Ling-Hong Hung
School of Engineering and Technology, University of Washington
Box 358426, Tacoma, WA 98402, USA

Evan Straw
University of Washington
Seattle, WA 98195, USA

Shishir Reddy
School of Engineering and Technology, University of Washington
Box 358426, Tacoma, WA 98402, USA

Zachary Colburn
Biodepot LLC
Seattle, WA 98195, USA

Ka Yee Yeung*
School of Engineering and Technology, University of Washington
Box 358426, Tacoma, WA 98402, USA

October 22, 2021

*Corresponding author: kayee@uw.edu

Abstract

Summary: Biomedical image analyses can require many steps processing different types of data. Analysis of increasingly large data sets often exceeds the capacity of local computational resources. We present an easy-to-use and modular cloud platform that allows biomedical researchers to reproducibly execute and share complex analytical workflows to process large image datasets. The workflows and the platform are encapsulated in software containers to ensure reproducibility and facilitate installation of even the most complicated workflows. The platform is both graphical and interactive allowing users to use the viewer of their choice to adjust the image pre-processing and analysis steps to iteratively improve the final results. We demonstrate the utility of our platform via two use cases in focal adhesion and 3D imaging analyses. In particular, our focal adhesion workflow demonstrates integration of Fiji with Jupyter Notebooks. Our 3D imaging use case applies Fiji/BigStitcher to big datasets on the cloud. The accessibility and modularity of the cloud platform democratizes the application and development of complex image analysis workflows.

1 Introduction

Analysis of biomedical big data often requires complicated and resource intensive analytical workflows consisting of different components. Software containers encapsulate executables and scripts enabling reproducibility and facilitating execution of complicated multi-step workflows using resources available on the cloud. The isolation afforded through the use of containers allows scientists to specialize and develop modules for the analytical steps that fit their research interests. For example, in the case of genomics data, most analyses are performed using open source modules from many different research groups that are connected using scripts and/or specialized workflow languages.

Early computational methods for quantifying and statistically analyzing microscopy images were limited. For example, in the 1980s, several methods for quantifying cell areas required crude approximations Warth (1989) or the division of photographic images into many polygons so their areas could be calculated and summed (Bates and Cheng, 1988). At that time, the gold standard for calculating cell area was based on physically cutting out and weighing developed film (Bates and Cheng, 1988). As microscopy data became digital, several companies began offering software with expensive price tags to enable these analyses. Custom made imaging analysis tools were also developed to satisfy niche applications. Today, a mix of open source and proprietary software cover the imaging analysis landscape. An example is Fiji (Schindelin *et al.*, 2012), a widely used and open-source package which extends the basic image processing capabilities of ImageJ (Schneider *et al.*, 2012) with plugins and macros. Fiji has extensive plugins, comprehensive documentation, and is extensible.

In contrast to genomics data, most image processing software is monolithic rather than modular and often tightly linked to the equipment used to generate the data. Omnibus software packages such as Fiji (Schindelin *et al.*, 2012), MetaMorph, and Image Pro (O'Mahony *et al.*, 2005) can be convenient but lack flexibility when the desired analytical function is not present. Users frequently need to manually link together image processing, segmentation, and statistical analysis tools to complete their analytic workflows. In addition, components that are part of large packages are particularly susceptible to the negative, and unexpected repercussions of software updates, costing these users time and reducing the reproducibility of their work. Using software containers to isolate individual components and link them together module via simple APIs simplifies installation, increases reproducibility (Mattonen *et al.*, 2020) and encourages software contributions from the research community.

Moreover, modern day bioimage analysis is not restricted to image processing specific software and can include statistical and visualization software, or be integrated with genomics data analysis as in spatial genomics. In addition, computationally demanding image processing, such as light sheet microscopy, often consists of gigabyte or even terabyte-sized datasets which can greatly benefit from on-demand computing resources on the cloud. It is challenging to set up interactive cloud graphics and many software tools provide a headless mode to circumvent this problem. However, the full potential of these tools is *not* realized without access to

the interactive image processing components.

We introduce the Biodepot-workflow-builder (Bwb) platform as a solution to address these difficulties. Bwb is an interactive graphical platform that allows non-computer experts to create and execute their analytical workflows locally or on the cloud (Hung *et al.*, 2019). Bwb is entirely containerized allowing autonomous modules from different academic or commercial sources to be reproducibly linked together to form an analytical pipeline. Unlike other workflow builders, Bwb runs workflows inside a container with graphical desktop that supports interactive graphical applications such as Jupyter notebooks (Kluyver *et al.*, 2016). Bwb has been used for analysis of multi-omics data but the native support for interactive and containerized graphics packages makes it particularly well-suited for analysis of image data. In this work, we demonstrate the utility of Bwb in a multi-step, multi-container, and containerized analysis of focal adhesions using a download module, a Fiji macro for segmentation, and Jupyter notebooks for data analysis and visualization. We also demonstrate the easy deployment of Fiji for 3D image analysis running the computationally demanding BigStitcher (Hörl *et al.*, 2019) plugin on the cloud, while retaining the interactivity and functionality of Fiji executed on a local computer.

2 Approach

The major technical challenge in using Fiji on the cloud, is to support the same graphical interface and display that one would see on a laptop. We have previously described two methodologies for accomplishing this using software containers. One is to export the graphics drawing (X11) commands to the client computer (Hung *et al.*, 2016). This requires software on the client to process these commands and draw the screen. The other approach is to draw the screen and then export the resulting frames using a VNC protocol (Mittal *et al.*, 2017). This also requires VNC viewer software on the client to manage the transfer and buffering of the frame data. The Bwb uses both methodologies. A container in a workflow exports X11 commands to the Bwb desktop environment which draws the graphics screens. The Bwb then exports the screen to a VNC viewer that displays the screen and manages user interactions. Since both the workflow and Bwb are encapsulated in containers, functioning on the cloud only requires that additional ports be opened to allow for communication of VNC data between the server and the client. Finally, Bwb is compatible with the browser-based noVNC client which allows users to use their browser to directly interact with the cloud workflow and avoid the need to install additional VNC software. As a result, the Bwb only requires Docker as a dependency and auto-downloads and installs itself. To create a Fiji module or widget for Bwb we first constructed a Fiji Docker container and uploaded it in our Biodepot repository for public downloads. The rest of the module is built using Bwb's form based interface and can be easily changed and customized by the user if desired. This is saved as a Bwb workflow, which is a directory of JSON and XML files. To start Fiji or any other workflow, the user launches the Bwb application locally or

on the cloud and connects to it using a browser or VNC client. Using the Bwb menu, the user then navigates to the Fiji workflow to load it. A set of icons appear and double clicking on the Fiji icon then once on the start button will start Fiji. Bwb will automatically download or create the Fiji container if necessary and then Fiji will launch. Users can also load and mix and match different modules with Fiji to form new workflows using Bwb's drag and drop interface. Graphics and interactions proceed as they would if Fiji were launched locally on the laptop but through the browser window instead of a window on the desktop.

3 Results

We demonstrate the utility of our approach by analyzing focal adhesions and stitching three dimensional (3D) images.

3.1 Analysis of focal adhesions

Cell motility is crucial to many biological processes, including development, wound healing, and metastasis. It is driven by cell-substrate interactions, which are mediated by focal adhesions, large multi-protein complexes situated on the basal surface of cells and most often near the leading edge (De Pascalis and Etienne-Manneville, 2017). They indirectly link the actin cytoskeleton and extracellular matrix proteins. This enables them to translate actin bundle contraction into traction forces that pull them across a substrate (Schwarz and Gardel, 2012). Although all focal adhesions are composed of a core set of proteins, including vinculin and paxillin, there are significant compositional differences between cell types with over 100 distinct proteins having been reported to associate with focal adhesions (Winograd-Katz *et al.*, 2014). Moreover, the numbers, sizes, and shapes of focal adhesions differ between cell types and are influenced by substrate stiffness, extracellular matrix, and signaling molecules, with these metrics being associated with cell motility (Kim and Wirtz, 2013). Calculating the numbers, sizes, and shapes of focal adhesions requires the segmentation of these microscopic structures in immunofluorescent images. Since dozens or more focal adhesions can be present in each cell, this is a very laborious process. There is therefore a significant, cross-discipline interest in methods to automate focal adhesion segmentation (Berginski and Gomez, 2013). However, because of differences between cell types, immunostaining methods, and imaging platforms and settings, this is not an easily standardizable task. Flexible methods for modifying existing segmentation models or generating new ones are desperately needed.

The demo workflow implements segmentation of focal adhesions using the algorithm described by Horzum *et al.* (2014). A set of fluorescently labeled paxillin images available from the Focal Adhesion Analysis Server described by Horzum *et al.* (2014) was first downloaded using the Download Files widget in Bwb, along with the LoG3D plugin for Fiji by Sage *et al.* (2005), which implements a Laplacian of Gaussians filter. We

implemented an option in the Fiji widget to specify the directory containing the LoG3D plugin.

A containerized copy of Fiji was then called to execute an ImageJ macro that triggers actions to perform the image processing steps given by Horzum *et al.* (2014) that isolate the focal adhesions in the image. The macro then used the built-in "Analyze Particle" plugin to compute and save the area (in pixels^2) and centroid coordinates for each focal adhesion to a CSV file. An ellipse was also fitted to each focal adhesion and the lengths of the major and minor axes of the ellipse as well as the angle of the major axis with respect to the horizontal were saved. Since ImageJ's macro language is intended to execute graphical user interface commands, the graphics and interface were exported from the Fiji container so the user can observe the status of the segmentation and interactively adjust the image in real time; see Figure 1 (B).

Another macro was then executed in Fiji to identify the outline of the cell in each time-slice of the image; this is important to determine the orientation of each focal adhesion relative to the edge of the cell. The macro first performed automatic thresholding, and then performed a series of morphological operations to demarcate the cell. The operations consisted of morphological dilation, closure, filling holes using the Fiji "Fill Hole" plugin, and erosion to return the cell to approximately its original shape. It was determined interactively, that three rounds of the morphological operations were required to close holes and remove noise. The "Outline" operation in Fiji was then used to find the cell borders, which was saved as xy-coordinate pairs to a text file using the "Save XY Coordinates..." command.

Finally, a Jupyter notebook was executed to analyze and visualize the data from the previous steps. The Pandas (McKinney *et al.*, 2010) and NumPy (Harris *et al.*, 2020) libraries were used to read the CSV file and obtain the segmentation and border coordinates. The Matplotlib (Hunter, 2007) library was then used to create histograms to visualize the distribution of areas, aspect ratios (i.e. the ratio of the lengths of the major and minor axes), and the angles relative to the cell edge observed in the focal adhesions. Two separate widgets are used. The first widget loads all the required libraries and automatically executes the analysis to generate the final graphs. The second widget displays a copy of the executed notebook with the generated results. Either notebook can be altered which gives the user the ability to interactively explore different visualization and analysis options for a particular dataset and then decide whether to commit them to the automated part of the workflow to be applied to other datasets. See Figure 1 (A) for a screenshot of the complete workflow running in the Bwb platform and Figure 1 (B) for graphical output of this workflow. A video demonstration of this workflow is available at `https://youtu.be/31dJ-KP4V2g`.

3.2 Stitching of 3D images

Advances in microscopy techniques, such as lightsheet or confocal microscopy, have led to the generation of large overlapping three dimensional (3D) images. However, the raw data acquired with the microscope are not

directly suitable for visualization and analysis. Digital reconstruction of these 3D images require stitching and fusion of large numbers of overlapping image tiles. Additionally, these 3D datasets can be large, particularly for lightsheet microscopy which is often on the terabyte (TB) scale. This leads to computational challenges in terms of storage, efficient analysis, and visualization.

BigStitcher is a software package that enables interactive visualization, efficient image alignment and deconvolution of multi-tile and multi-angle image datasets (Hörl *et al.*, 2019). Both alignment and viewing of terabyte-size datasets composed of overlapping three-dimensional (3D) image tiles are supported. BigStitcher offers options for fully automatic or interactive stitching. BigDataViewer (Pietzsch *et al.*, 2015) is provided with the package, for visualization of the aligned datasets. To accommodate the computational requirements, the original benchmarks for BigStitcher in (Hörl *et al.*, 2019) were performed using high-end servers. However, many researchers do not have access to this type of locally dedicated infrastructure but do have access to pay-as-you-go resources on the public cloud. We demonstrate how enabling execution of Fiji on the cloud democratizes access to computationally intensive applications such as BigStitcher.

BigStitcher is installed as a plugin through the Bwb Fiji widget. The demo workflow uses BigStitcher to align and display a multi-tile dataset of a 3D confocal scan of the nervous system of a Drosophila larva. The images consist of 6 tiles and 3 channels each. This data is available as an example dataset within BigStitcher's Fiji plugin documentation and serves as a good example for the cloud capabilities of the software package. After opening the raw data in BigStitcher, the filename patterns for channels and tiles are selected. The images are then pulled up in the BigDataViewer plugin and aligned to a regular grid for easier viewing. Manual alignment is skipped in this workflow but is available as an option for more complex datasets that require user intervention in stitching. Figure 1 (C) shows the stitched image from the Drosophila dataset (123 MB) displayed in the BigDataViewer plugin. Stitching and viewing the raw data is completed in 20 seconds on a m5dn.4xlarge AWS EC2 instance. A video demonstration of this workflow is available at `https://youtu.be/lUDed8AW0z8`.

4 Conclusions

We have demonstrated the utility of Bwb, a containerized, graphical cloud workflow execution platform for image analyses in two separate use cases. The Fiji image analysis suite has been containerized for reproducible deployment on local and cloud servers and functions in the same interactive manner it does on a local host. We have connected Fiji to separate modules for downloading data, and analyzing and visualizing the results using Jupyter notebooks. We showed how the containerization enables execution of computationally expensive applications such as BigStitcher on the cloud. The modularity, reproducibility and accessibility of this approach democratizes both the process of image analysis and development of new analytical methods and packages. Finally, Bwb also supports many ge-

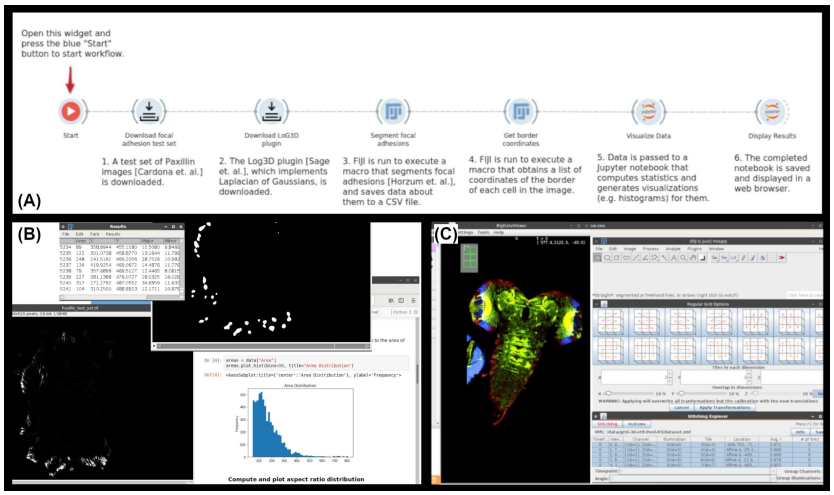

Figure 1: (A) The workflow for segmenting focal adhesions. Execution begins at the play button icon and proceeds to the right, with each widget being triggered upon completion of the task performed by the one to its left. (B) The workflow in (A) in action, with focal adhesion segmentation being performed in Fiji on the top and left, and results being displayed in Jupyter on the right. The top image window and table specifically show the results output by the "Analyze Particles" command; each white blob represents a particle, with statistics for each being shown in the table. (C) The left half of the image contains the stitched Drosophila nervous system displayed using BigDataViewer within the BigStitcher plugin. The right half of the image shows options for alignment to different regular grids as well as information for each tile separately.

nomics workflows that can potentially be connected with image software like Fiji for interactive, cloud enabled spatial genomics analysis.

Acknowledgements

LHH, SR and KYY are supported by NIH grant R01GM126019. LHH, ES and ZC are supported by NCI SBIR contract 75N91021C00022. We would also like to thank Amazon Web Services for cloud credits. The content is solely the responsibility of the authors, and does not necessarily represent the official views of the National Institutes of Health.

LHH and KYY also have equity interest in Biodepot LLC, which receives compensation from NCI SBIR contract numbers 75N91020C00009 and 75N91021C00022.

References

Bates, A. and Cheng, H. (1988). Bullous keratopathy: a study of endothelial cell morphology in patients undergoing cataract surgery. *British journal of ophthalmology*, **72**(6), 409–412.

Berginski, M. E. and Gomez, S. M. (2013). The focal adhesion analysis server: a web tool for analyzing focal adhesion dynamics. *F1000Research*, **2**.

De Pascalis, C. and Etienne-Manneville, S. (2017). Single and collective cell migration: the mechanics of adhesions. *Molecular biology of the cell*, **28**(14), 1833–1846.

Harris, C. R. *et al.* (2020). Array programming with NumPy. *Nature*, **585**(7825), 357–362.

Hörl, D. *et al.* (2019). Bigstitcher: reconstructing high-resolution image datasets of cleared and expanded samples. *Nature Methods*, **16**(9), 870–874.

Horzum, U. *et al.* (2014). Step-by-step quantitative analysis of focal adhesions. *MethodsX*, **1**, 56–59.

Hung, L.-H. *et al.* (2016). Guidock: using docker containers with a common graphics user interface to address the reproducibility of research. *PloS one*, **11**(4), e0152686.

Hung, L.-H. *et al.* (2019). Building containerized workflows using the biodepot-workflow-builder (bwb). *Cell Systems*, **9**, 508–514.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, **9**(3), 90–95.

Kim, D.-H. and Wirtz, D. (2013). Focal adhesion size uniquely predicts cell migration. *The FASEB Journal*, **27**(4), 1351–1361.

Kluyver, T. *et al.* (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.

Mattonen, S. A. *et al.* (2020). Quantitative imaging feature pipeline: a web-based tool for utilizing, sharing, and building image-processing pipelines. *Journal of Medical Imaging*, **7**(4), 042803.

McKinney, W. *et al.* (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Mittal, V. *et al.* (2017). Guidock-VNC: Using a graphical desktop sharing system to provide a browser-based interface for containerized software. *Gigascience*, **6**(4), giw013.

O'Mahony, R. *et al.* (2005). Comparison of image analysis software packages in the assessment of adhesion of microorganisms to mucosal epithelium using confocal laser scanning microscopy. *Journal of microbiological methods*, **61**(1), 105–126.

Pietzsch, T. *et al.* (2015). Bigdataviewer: visualization and processing for large image data sets. *Nature methods*, **12**(6), 481–483.

Sage, D. *et al.* (2005). Automatic tracking of individual fluorescence particles: Application to the study of chromosome dynamics. *IEEE Transactions on Image Processing*, **14**(9), 1372–1383.

Schindelin, J. *et al.* (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, **9**(7), 676–682.

Schneider, C. A. *et al.* (2012). Nih image to imagej: 25 years of image analysis. *Nature methods*, **9**(7), 671–675.

Schwarz, U. S. and Gardel, M. L. (2012). United we stand–integrating the actin cytoskeleton and cell–matrix adhesions in cellular mechanotransduction. *Journal of cell science*, **125**(13), 3051–3060.

Warth, A. D. (1989). Relationships among cell size, membrane permeability, and preservative resistance in yeast species. *Applied and environmental microbiology*, **55**(11), 2995–2999.

Winograd-Katz, S. E. *et al.* (2014). The integrin adhesome: from genes and proteins to human disease. *Nature reviews Molecular cell biology*, **15**(4), 273–288.