MiPipeline (Microscopy Pipeline): A User Friendly Software Environment for Microscopy Image Analysis and Informatics

Kaustav Nandy[1], Yanling Liu[2], David Mott[2], Karen Meaburn[3], Tom Misteli[3], Stephen J. Lockett[1] and Prabhakar R. Gudla[1]

[1] Optical Microscopy and Analysis Laboratory, Leidos Biomedical Research Inc., Frederick National Lab, Frederick MD, USA
[2] Advanced Biomedical Computing Center, Leidos Biomedical Research Inc., Frederick National Lab, Frederick MD, USA
[3] Cell Biology of Genomes, National Cancer Institute, NIH, Bethesda MD, USA

***Abstract*—Recent advances in the field of optical microscopy have enabled scientists to image complex biological processes across a wide range of spatial and temporal resolutions, resulting in an exponential increase in optical microscopy data. There is now a need for a computing environment that provides accurate, high-throughput image analysis, in conjunction with data provenance, easily accessible workflow sharing, information visualization and analysis. We report the development of MiPipeline, an environment that provides the aforementioned capabilities. A case study involving human breast cancer detection using tissue micro-array image data illustrates the capabilities of the environment. An 80 fold reduction in computation time was achieved using MiPipeline workflow.**

***Index Terms*—visual programming, workflow, high performance computing, information visualization**

I. Introduction

The past few years have witnessed unprecedented improvements in optical microscopes particularly in terms of spatial resolution and automated generation of high-content imaging data, inevitably leading to an explosion in the quantity of acquired data (dataset sizes vary, are several TBytes in some cases). Concomitantly, there is increasing need for rigorous quantification of complex biological interactions often at hierarchical scales (tissues, cells, sub-cellular/nuclear, and molecular) that is represented in these images. Thus, analysis and visualization of large, information-rich bio-image datasets is becoming increasingly important and challenging. Since, manual processing and interpretation of such datasets is impractical and subjective, automatic analysis is essential. However, the variations in the implementations of image processing algorithms and the adaptation of analysis procedures due to differences in samples and/or acquisition platforms lead to results that are poorly reproducible. This brings forth with it some urgency for data and process provenance.

To address the analysis needs of large microscopic bio-image datasets, an ideal comprehensive computational environment should provide: access to algorithms and libraries from multiple disciplines (image processing, computer vision, machine learning, and bio-statistics), data and process provenance for accurate tracking of data and analysis procedures, a collaborative workflow management so that multiple participants can seamlessly contribute to data analysis and development of analysis tools, an environment for visual programming requiring minimal programming knowledge, integration with high performance computing (HPC) clusters, and information visualization and analytics.

Several promising data processing platforms for bio-image data have emerged in the past few years that each address a subset of these needs. They include: OME/OMERO [1], Farsight Toolkit [2], Cell Profiler[3] and ICY [4] which provides access to several algorithms and libraries for analysis of microscopic data. From a workflow management standpoint, several additional open-source and commercial platforms called workflow management systems (WMS) exist, e.g., KNIME[5], TAVERNA[6], KEPLER[7], LONI-Pipeline environment [8], PSOM [9], Galaxy [10] and Pipeline Pilot (Accelrys, Inc., San Diego, CA, USA). These platforms provide access to algorithms from a wide-array of disciplines and can be executed on existing HPCs. Most of them also provide visual data processing and reporting.

The goal of this study was to provide a comprehensive computing environment for analysis of biological samples imaged with optical microscopy. The environment, "MiPipeline", is based on the LONI Pipeline [8] for image processing and analysis coupled with a new web browser based visual analytics. The environment provides: a collaborative environment with independence from specific programming API (e.g., Java or Python), end-user-friendly visual workflows, tight integration with existing HPC resources, data provenance and reproducibility through provision of an XML [11] backbone that tracks image data and its progress through processing and analysis algorithms, rapid transition from prototyping to production, availability of Bioformats for importing images in formats commonly used in biomicroscopy and ImageJ/Fiji for image processing and analysis, preexisting pool of multi-disciplinary algorithms included in LONI that can be used in creating visual workflows, seamless information exploration and analytics via a web interface. The main contribution of this work is the development of Matlab based tissue analysis modules which were integrated with and are available for the first time on the LONI Pipeline server for processing optical microscopy tissue datasets, integration of popular microscopic image analysis tools such as ImageJ and BioFormats and the development of an integrated information visualization framework which uses an XML based data backbone developed specifically to organize inherent hierarchical data from optical microscopy images.

II. MiPipeline Environment

Figure 1 illustrates the four components of the MiPipeline environment along with their interactions, namely, 'USER

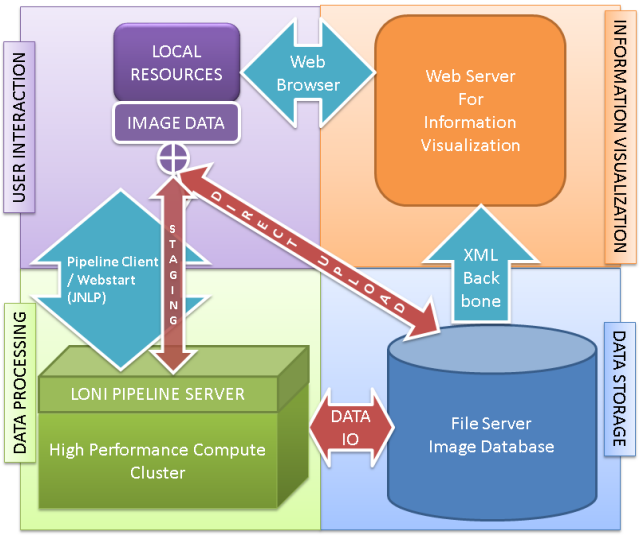

Fig.1. The MiPipeline environment. As novel contributions, MATLAB based tissue analysis modules were deployed for the first time on the LONI Pipeline (DATA PROCESSING) along with the integrated web-based INFORMATION VISUALIZATION component based on an XML based data provenance backbone.

INTERACTION', 'DATA PROCESSING', 'DATA STORAGE' and 'INFORMATION VISUALIZATION'.

The first component, 'USER INTERACTION' includes organizing and transferring images of biological samples, software development activities if needed for custom applications, setting up the processing pipeline and post-analysis manual data mining with the interactive information visualization component: 'INFORMATION VISUALIZATION'.

The 'DATA PROCESSING' component is based on LONI-distributed pipeline server (DPS) and provides a user friendly interface (via LONI Pipeline Client) to high performance computing resources. This component gets the input data either directly from the user ('STAGING',Fig.1) or from a central file server ('DATA STORAGE'). On analysis completion, results are streamed back to the user ('STAGING') or saved on the central storage.

The 'DATA STORAGE' is a central data server accessible from the user local resources and compute nodes of the 'DATA PROCESSING' component. A user can directly upload ('DIRECT UPLOAD' in Fig. 1) their data onto the 'DATA STORAGE' which is subsequently processed by the 'DATA PROCESSING' component. A user can also create an XML information backbone, which is subsequently used for information visualization and data provenance.

The 'INFORMATION VISUALIZATION' component of the MiPipeline environment is web technology based and uses the XML backbone to visualize metadata and results for exploration and mining purposes.

A. LONI Pipeline

The LONI Pipeline tool is well suited for deployment of intricate and involved data analysis pipelines and warrant minimal maintenance and support overhead. Once a module/workflow is deployed on the main server, a user with no programming background can easily import their data and

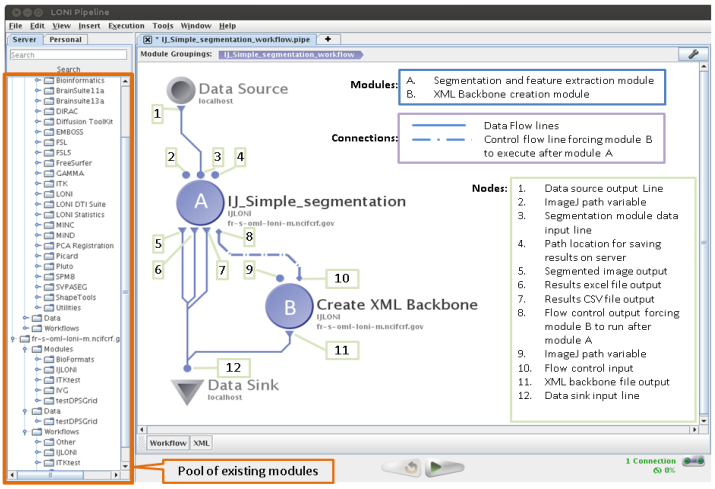

Fig.2. LONI Pipeline client showing a simple workflow implementing ImageJ's watershed algorithm based image segmentation and XML backbone creation. Pre-existing analysis modules are also highlighted in the red box..

process it using a rich set of pre-existing workflows or their own custom made workflows created by interconnecting the available server or local custom-made modules (see figure 2 for a simple example). Along with a large pool of central server based repositories catering for the needs of a wide user base, local repositories of custom-made modules, catering to the requirements of focused research groups, can both be maintained.

The visual programming environment offered by LONI Pipeline client is also conducive for rapid application prototyping, deployment and sharing. Once developed, generic as well as custom made workflows can not only be easily accessed on the server via the LONI pipeline client, but can also be shared as .pipe files which are XML documents encoding the workflow details.

LONI Pipeline is a versatile platform where users can easily incorporate and interconnect softwares developed in disparate languages as modules as long as they can be invoked from the command line prompt on the server. For example, a single pipeline can incorporate modules developed using R, Perl, Java, Python, C, C++, Octave or Matlab (using Matlab Runtime Component). The Pipeline server provides a seamless integration of the modules to existing high performance computing infrastructure available locally or at the main server.

The current deployment of MiPipeline 'DATA PROCESSING' module is hosted on LONI Pipeline Server at cranium.loni.usc.edu and provides (guest) access to LONI DPS installed on top of Oracle/Sun Grid engine (SGE). The main high performance computation (HPC) cluster include a Linux (CentOS 6.4, 64-bit) computer cluster with one head node and thousands of multicore compute nodes . Further details can be found at 'http://pipeline.loni.usc.edu/'.

B. MiPipeline Data Provenance

MiPipeline provides two layers of data provenance. The LONI Pipeline tool ('DATA PROCESSING') itself provides the first layer of module and workflow provenance in the form of internal XML [11] files. For additional data provenance in

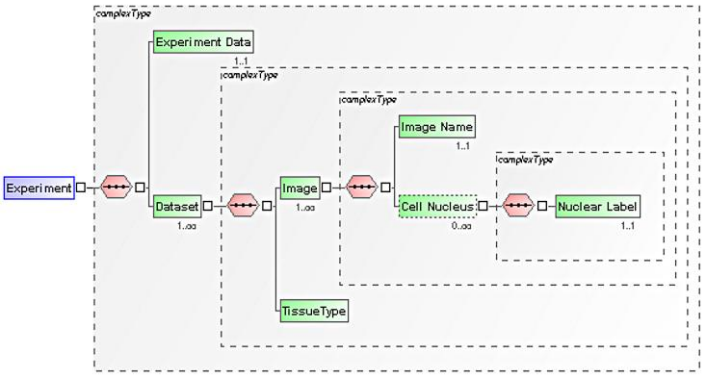

Fig. 3. PAGODA XML schema

MiPipeline, we have developed a second layer of application specific data provenance in the form of an XML information backbone which was specifically designed to capture inherently hierarchical data extracted from optical microscopy images from biological datasets. Such information is not captured in the first provenance layer offered by LONI Pipeline. The application specific backbone needs to be created by the user depending on the needs of the application. The XML is populated with information such as meta-data about the biological sample, steps taken in the image processing and results. A sample XML schema is displayed in figure 3 which illustrates the hierarchical structure of the XML file capturing multi-scale information starting at the dataset level and going down to the nuclear level. This information can be seamlessly visualized, explored and mined using the MiPipeline information visualization component.

C. MiPipeline Information Visualization

In the case of large image datasets, it is a challenge for users to explore, navigate and extract connections, trends, and distributions from the meta-data and experimental output. To address this need we have developed a custom web-browser-based flexible yet powerful dynamic information visualization framework. The goal was to provide straightforward high-level overview and interactive filtering mechanisms to quickly narrow down on to data points of interest. It was built using open source JavaScript visualization techniques (d3.js[12]) and NoSQL database (MongoDB (http://www.mongodb.org)). The front-end of this framework provides a web application for efficient data interaction and navigation, which can be accessed using any modern web browser. Data visualization was done with minimal overhead development by integrating multiple JavaScript libraries such as d3.js, jQuery(http://www.jquery.com), and Bootstrap (http://twitter.github.io/bootstrap/). In its most generic version, this system can render any XML file as an interactive hierarchical tree structure, in lines with the sample XML schema displayed in Figure 3, using d3.js functions.

For visualization and analysis purposes, XML metadata files were parsed and stored in MongoDB database in JavaScript Object Notation (JSON) format, which was required by d3.js for rendering. Because of the hierarchal structure of XML metadata, we have chosen tree structure view to render the high-level overview. The tree structure view is dynamically generated at run time and allows us to generate sub-tree structure created from search results for rendering. In the MiPipeline information visualization framework we added additional visual exploration and mining capabilities of image display, data filtering, feature visualization and searching. These are incorporated based on the nature of information to be visualized for custom applications. All capabilities can be explored at http://mipipeline.ncifcrf.gov/ivaan/.

Table 1. Modules currently available in MiPipeline environment

Modules	Platform
LOCI BioFormats	*Java*
ImageJ Plugins such as QuickPALM	
Watershed based 2D nuclear segmentation	*Matlab*
FISH foci segmentation	
PAGODA modules	

D. Microscopy Tool Integration

We integrated some popular and powerful microscopy tools into the MiPipeline environment: LOCI Bio-Formats [13] library and ImageJ/Fiji [14]. LOCI Bio-Formats library enables reading of metadata as well as image data of more than one hundred file types including proprietary microscopy image formats. ImageJ/Fiji provides a wide array of microscopy image analysis tools that can furthermore be used in a parallel computing infrastructure. Advanced image analysis workflows can be created by interconnecting ImageJ tools where each one is an ImageJ macro. The only restriction is that the macro must be launched from the command line prompt and therefore must be able to execute as a headless, batch mode operation. The integration of such Java based applications is seamless and can be easily accomplished by making the appropriate jar files available on the Pipeline server. Table 1 shows a list of modules that are currently available in the MiPipeline environment.

III. Parallel Genome Organization Diagnosis Software (PAGODA) Case Study

Recent research has shown that the spatial positioning of certain genes within the cell nucleus differs between normal and cancerous human breast tissue samples. This preferential localization of the genes with respect to the center of the cell nuclei can be exploited to detect breast cancer as shown by studies involving analysis of both cell culture models [15] and human tissue samples [16,19]. This case study builds on the aforementioned findings and here its capabilities are extended from a research level finding to a sophisticated and practical software tool by providing a visual workflow based framework for high-throughput data processing and seamless information visualization.

The experimental data used for testing PAGODA comprised of 43 human breast specimens (40 breast cancer and 3 non cancerous breast disease samples). *HES5* and *FRA2* DNA sequences were labeled by fluorescence *in situ* hybridization (FISH), imaged and pseudo-colored as red and green color channels, respectively [16]. The counter stained cell nuclei were imaged and assigned to the blue channel. The input to the PAGODA pipeline was approximately 5.1GB

(1703 2D RGB images of 1024x1024 pixels, 3 channels and 8-bits per pixel). A small portion of the TMA data is available on cranium.loni.usc.edu (TMA_DEMO) for testing and the current PAGODA workflow is configured to demonstrate its capabilities on this smaller dataset.

A. Methods

The analysis of tissue section images involved: background removal, wavelet-based nuclear edge enhancement, automatic cell nuclei segmentation, logistic regression based selection of accurately-segmented nuclei, detection of FISH labeled gene sequences and measurement of the gene locations (FISH signals) with respect to the nuclear center. The details of the algorithms were reported in [18,19] and individual modules for the algorithms are publicly available on cranium.usc.edu LONI Pipeline server. The novelty of the approach was in the fact that a pattern recognition engine automatically identified a subset of accurately segmented cell nuclei for further utilization for extraction of biologically significant results. Core PAGODA modules were implemented using MATLAB 2010b [17] making use of functions from commercial MATLAB toolboxes: Image Processing and Statistics toolboxes. The modules were converted to executables (using MATLAB compiler and required MATLAB Component Runtime for execution) with appropriate wrappers (BASH scripts) for accommodating custom input-output requirements. The aforementioned image analysis steps were offered as MiPipeline PAGODA modules accompanied by experimental metadata storage as an extensible markup language (XML) file. The adaptation of the MATLAB based modules onto the LONI Pipeline infrastructure made the application easily scalable and also publicly available without requiring a MATLAB license to run the application.

B. Information Visualization

Figure 3 shows the XML schema for creating the XML information backbone which was used for information visualization. The web-based interface provided real-time interaction with the experimental meta-data visualized as a tree, display of intermediate results and images, feature visualization of segmented cell nuclei, data filtering and mining capabilities. Sample XMLs can be visualized at http://mipipeline.ncifcrf.gov/ivaan/ for exploring additional capabilities.

C. Results

The validation and accuracy of the segmentation module and machine learning modules of PAGODA have been reported previously [18,19]. Using MiPipeline PAGODA, the data was compared to a manually pooled benchmark dataset using Kolmogorov-Smirnov (KS) test with a 1% level of significance. For datasets where sufficient number of nuclei (>=75 which was shown to provide statistically significant results [16]) were identified, the centrality measure of HES5 and FRA2 were able to correctly predict the patient tissue type (whether cancer or normal) correctly in 86.8% and 92.1% cases. However, it should be noted that due to the low number of non-cancerous breast disease tissue samples, the mentioned accuracy parameters reflect the sensitivity of the study and fail to measure the specificity of the test.

On average, the processing time for each image from a patient sample was approximately 7-8 mins and the entire analysis was completed in ~2 hours on the USC-LONI's Cranium cluster, resulting in a 80-fold speedup compared to the sequential version of the code. The speedup was sublinear, since not all the modules and operations in PAGODA processing were parallelized. The analysis of the input data using MiPipeline PAGODA, resulted in a raw output occupying ~35 GB in disk space, including redundant intermediate files from each module in the workflow. The XML information backbone file generated from the full analysis was, however, only 32 MB and had appropriate network locations for intermediate results on the network file server.

IV. Conclusions and Future Work

We report the development of a visual programming based high throughput analysis and integrated information visualization environment called MiPipeline for handling large microscopy image datasets. The central idea behind the development of MiPipeline was to provide user friendly access to specialized high performance computing infrastructure via the LONI Pipeline client requiring minimal programming knowledge along with an integrated information visualization platform. MiPipeline has provisions for a comprehensive process and data provenance via XML backbones that are further utilized for information visualization. The web-based MiPipeline information visualization module utilized javascript based libraries to visualize meta data and experimental results. A number of tools which are very useful in handling microscopy based datasets such as LOCI- Bioformats and ImageJ have been ported to the MiPipeline environment and future work will concentrate on broadening the array of modules that are available on MiPipeline. Moreover, several improvements are planned for the information visualization framework to make it more generic and intuitive by incorporating other information visualization and mining tools to ease understanding and interpretation of underlying data trends and results.

Acknowledgments

This project has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institutes of Health, under Contract No. HHSN261200800001E, in part by the Intramural Research Program of the National Institutes of Health, National Cancer Institute, Center for Cancer Research and by a Department of Defense Breast Cancer Idea Award.. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.The Office of Human Subjects Research (OHSR) at the National Institutes of Health, USA determined on January 2 2008 that federal regulations for the protection of human subjects do not apply to this research project. The human material used in this study had been de-identified before any of the authors received it. Fluorescence imaging was performed at the National Cancer Institute Fluorescent Imaging Facility, Bethesda, MD, USA. We are also thankful to the LONI Pipeline team at University of Southern California, especially Petros Petrosyan for hosting PAGODA and providing excellent support.

References

[1]. C Allan et al., OMERO:flexible, model-driven data management for experimental biology. Nature Methods, 9:245–253, 2012.

[2]. FARSIGHT Toolkit, http:// www.farsight-toolkit.org

[3]. Anne Carpenter et al., CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biology, 7(10):R100, 2006.

[4]. F. de Chaumont et al., Icy:An open bioimage informatics platform for extended reproducible research, Nature Methods,9:690–696, 2012.

[5]. Michael R. Berthold et al., KNIME: The Konstanz Information Miner. In Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). Springer, 2007.
[6]. K. Wolstencroft et al., The Taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. Nucleic Acids Research, 41(W1):W557–W561, 2013.
[7]. Bertram Ludascher et al., Scientific workflow management and the Kepler system: Research articles. Concurr. Comput. :Pract. Exper., 18(10):1039–1065, August 2006
[8]. Ivo Dinov et al., Neuroimaging Study Designs, Computational Analyses and Data Provenance using the LONI Pipeline. PLoS ONE, 5(9):e13070, 09 2010.
[9]. Pierre Bellec et al., The pipeline system for Octave and Matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows. Frontiers in Neuroinformatics, 6(7), 2012.
[10]. Daniel Blankenberg et al., Galaxy: A web-based genome analysis tool for experimentalists. Current protocols in molecular biology, pages 19–10, 2010.
[11]. T. Bray, J. Paoli, and C.M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. W3C Recommendation, Feb. 1998.
[12]. M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven Documents. Visualization and Computer Graphics, IEEE Transactions on, 17(12):2301 – 2309, Dec. 2011.
[13]. Melissa Linkert et al., Metadata matters: access to image data in the real world. The Journal of Cell Biology, 189(5):777–782, 2010.
[14]. C.A. Schneider, W.S. Rasband, and K.W Eliceiri. NIH Image to ImageJ:25 years of image analysis. Nature Methods, 9:671–675, 2012.
[15]. K. J. Meaburn and T. Misteli. Locus-specific and activity-independent gene repositioning during early tumorigenesis. J. Cell Biol., 180(1):39–50, 2008;.
[16]. K. J. Meaburn, P. R. Gudla, S. Khan, S. J. Lockett, and T. Misteli. Diseasespecific gene repositioning in breast cancer. J Cell Biol., 187(6):801–812, 2009;.
[17].Matlab, release 2010B, 2010. http://www.mathworks.com.
[18]. W. Cukierski et al., Ranked retrieval of segmented nuclei for objective assessment of cancer gene repositioning. BMC Bioinformatics,13(1):232, 2012.
[19]. K Nandy, P.R. Gudla, R. Amundsen, K.J. Meaburn, T. Misteli, and S.J. Lockett. Automatic segmentation and supervised learning-based selection of nuclei in cancer tissue images. Cytometry A., 81(9):743–754, 2012.