Constructing a Modular Ancient DNA Pipeline using WDL

Tra-Vaughn M.C. James
Earlham College
Richmond, Indiana
tjames19@earlham.edu

ABSTRACT

Bioinformatics is an interdisciplinary field between biology, computer science and statistics, in which software is able to augment, and analyze biological data. To convert this data into useful information the use of various tools, parameters, and dynamically changing reference data is required [12]. As a result, workflow management system such as Shakemake and WDL were created to develop workflows that are scalable, repeatable and shareable. Such tools have become pivotal within large-scale labs, saving bioinformaticians time, resources and streamlining the analysis process. Bioinformatic genomics labs, specifically those involved in ancient DNA, also use a variety of tools and programs toward assessing and producing readable data. However, the scripts that pipeline and assemble such workflows are often not efficiently written. Moreover, there are slight discrepancies in the tools that particular labs use, differing from one to the other, this makes it more difficult to share and scale workflows from other labs. I propose constructing a modular WDL pipeline that will increase both the efficiency and usability, and share-ability of current ancient DNA workflows.

KEYWORDS

Bioinformatics, workflow, workflow manager, workflow management, workflow management tool, OpenWDL, WDL, Workflow Description Language, Genomics, Ancient, Ancient DNA, Pipeline Builder

ACM Reference Format:
Tra-Vaughn M.C. James. 2022. Constructing a Modular Ancient DNA Pipeline using WDL . In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Today, Bioinformatics is an evolving field, in which computing resources have become more powerful, readily available and workflows have increased in complexity. New workflow management tools (WMT) attempt to develop software that fully harnesses this computational power, creating intuitive implementations utilizing machine learning techniques. This streamlines the design of complex workflows. Such WMTs prove especially useful in ancient DNA research, in which current pipe-lining techniques prove inefficient toward producing reliable results. The major problems of these workflows stem from them being bespoke in nature, in which some tools utilized by one lab differ from another, this often creates issue when attempting to share or scale workflows from one another. Moreover, the overall structure of many of these workflows are inefficient, many often being hastily put together bash scripts, given current WMT capabilities, there are more efficient methods that make it easier to both edit and develop original workflows with new data and tools. Utilizing WDL and its GUI tool Pipeline Builder I seek to create an ancient DNA dating Pipeline Workflow. This pipeline will be modular in design and allow the user to utilize and string together a variety of tools that they can use to create data specific workflows. Due to the interface of the pipeline the user will also be able to easily configure input files, commands, and specify outputs.

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

2.1 OpenWDL

Originally created by the Broad Institute for the purposes of genome analysis pipelines, OpenWDL, a Workflow Description Language (WDL), has evolved to much more than its original intent. It provides a "way to specify data processing workflows with a human-readable and -writeable syntax. It makes it straightforward to define analysis tasks, chain them together in workflows, and parallelize their execution."[4] WDL requires an execution engine to run, being compatible with engines such as Cromwell, MiniWDL and dxWDL. It further supports Python, JavaScript, and Java. Due to this and OpenWDL becoming an open source community, it has become the basis for a slew of other WMT's, workflows, as well as an inspiration for new WMT implementations. The various WMT's utilizing OpenWDL, coupled with its familiarity and widespread use within the field, makes OpenWDL a prime candidate for the basis of my own implementation.

Cromwell is a "workflow execution engine that simplifies the orchestration of computing tasks needed for genomics analysis" [5], used to execute WDL scripts. It is now an open-sourced project, but was also originally developed by the Broad Institute. Cromwell can run on a variety of systems, including local and cloud-based services e.g. AWS (Amazon Web Services). This capability is shown through the use of the Terra platform, which can run WDL scripts, Terra includes a built-in Cromwell server that "interprets your WDL workflow script, transforms it into batches of individual analysis instructions, and finally dispatches it to a Google Cloud service called Pipelines API for execution" [2]. One of the most powerful features of Cromwell with WDL is the ability to create sub-workflows. This allows the "execution of an entire workflow as a step in a larger workflow, when a workflow calls another workflow, that second workflow is called a sub-workflow." [1] These sub-workflows can

contain within themselves other sub-workflows, thus, allowing for workflow nesting, which aids in workflow reuse. [6]

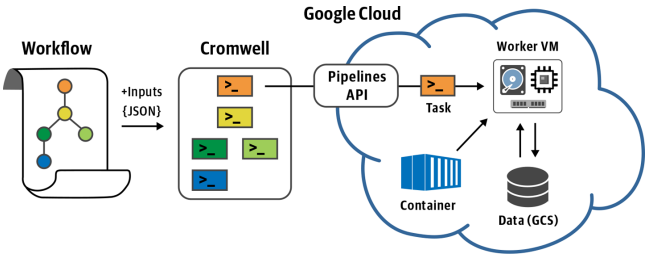

Figure 1: An overview of Cromwell with Terra. (https://3qim1p3jo50i2s787c216eqb-wpengine.netdna-ssl.com/wp-content/uploads/2020/11/cromwell-gcp-overview.png).

3 DESIGN AND IMPLEMENTATION

3.1 Workflow Design

Each workflow follow a similar schema to the process shown in the Data Architecture diagram bellow: First, we start of with Raw fastq files these are arranged by the location in which the sample was taken. The fastq files are then used as input for the extracting pipeline in which we join the left and right reads, perform a Quality test on them, and finally we filter the alignment's using Bowtie 2. The final output of this are Sequence Alignment Maps, in which are categorized by sample spot, read id, species id and k value (specifies first, second, or third best read). The second major part of the workflow is assembling a species of interest file, this a drive sheet of all the species that is of significant importance to the lab. The species of interest file along with the Sequence Alignment Maps are then used as input into the NCBI database, this will reveal if the DNA samples had any matches to the labs species of interest. The final output will Dating Gross indicating the species matches. Overall the current workflow used by the various labs are mostly multiple nested bash scripts. Bash, and almost any other command line tool can interface with WDL, thus, translating the workflow into WDL was as simple as taking the predefined commands placing them in WDL. However, the overall structure of these script still had various inefficiencies such as, hard coded inputs and outputs and the use of repetitive commands. Such made it difficult to change or modify the workflow for different values as each script, must be edited individually. Fortunately, as a means to address this issue, WDL allows you you specify the inputs and outputs needed, using an inputs.json file, in which allows me to reference such inputs directly throughout the script without placing them explicitly within it. Subsequently it also allows the user to change various inputs without interacting with the script directly. Moreover, through the use for loops and various tool WDL provides I was able to take repetitive and redundant code make it more efficient allowing for a more readable workflow. Essentially I was able to take entire separate bash files and turn them into short but efficient tasks within a single WDL script.

3.2 Workflow Operation

When operating the workflow, a user would edit an inputs.json file and fill in the required information, such as the working, input, and output directories, specific parameters for commands, and other such information. After such values have been submitted into the inputs file, using the Cromwell version of there choice, the user would finally run the workflow at the command line. When such is run the job is automatically dispatched to the slurm scheduler on the Earlham CS clusters. Each task within the workflow will be transcribed into a bash script by the Cromwell execution engine and will run in accordance to the inputs it is dependent on.The exception to this is the setup task, it will always run first to ensure that all the inputs submitted are valid. The subsequent tasks will follow only if their input parameters are met. For instance if a task is dependent on another tasks output it will of course be scheduled to run after that output is produced. As Cromwell is in control of when and what tasks will be run it will most often choose the most efficient method for scheduling tasks. For instance if two tasks have all inputs met and are ready to run they may be executed simultaneously. This function assists in making the workflow more efficient in use.

4 WHOLE-GENOME DATA ARCHITECTURE

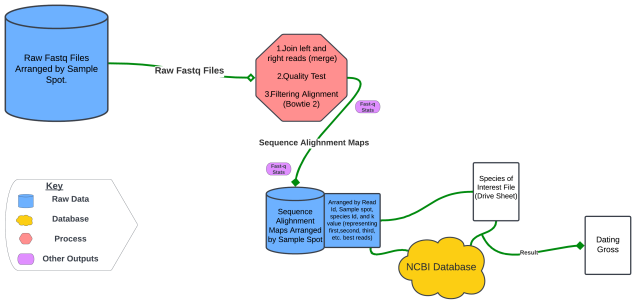

Figure 2: An overview of of how data follows throughout the whole genome Workflow.

5 16S DATA ARCHITECTURE

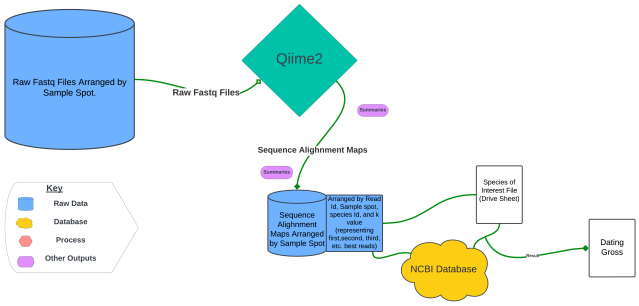

Figure 3: An overview of of how data follows throughout the 16s Workflow.

6 INTER-CONNECTIVITY OF LAB PROCESSES

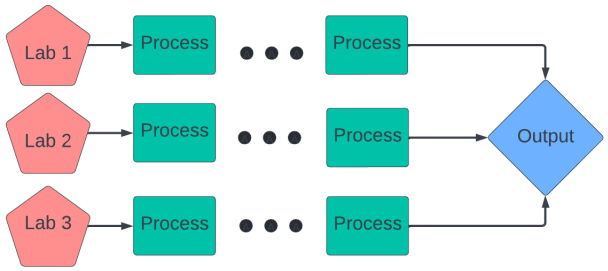

Figure 4: How each lab is interconnected in which they produce a common output.

7 RESULTS

I have successfully created WDL workflows for the Icelandic whole genome lab and the 16s mRNA lab. Overall, while incredibly useful when chaining tasks and outputs together, WDL has proved to be a somewhat difficult language to use and understand. This is primarily due to the poor placement of core information as to its operation and a lack of components in which would make it simpler to use. I especially found the former to be source of many of WDL's down falls. Simple operations such as using loops was not fully covered in the current documentation. Moreover, Pipeline Builder a GUI interface documented on the Terra wdl website as being used to manage and create WDL workflows was no longer being updated and maintained. This tool was to be used as a means to introduce novices to WDL workflow construction and management, but such could not be installed. Lastly, while not directly a cause of WDL's construction but more so due to the complexity and intricacy of Ancient DNA dating, it is difficult to run individual processes given how interdependent each process is to another. This unfortunately meant that a user would have to create another WDL script to run a specific process separate from the workflow, instead of centralized WDL script in which could handle both running individual tasks or the entire workflow. Therefore hindering how reproducible certain workflows are. Nonetheless, WDL has still dramatically improve the workflows for both the 16s and whole-genome labs when compared to previous methods used to manage them. Outputs are automatically configured and inputs are able to be globally declared, many tasks are able to run simultaneously, and the entire workflow can be scheduled via Cromwell to the Earlham's cluster slurm scheduler. Moreover, because the setup for WDL is so simple, workflows are able to be maintained, reproduced, shared and scaled in a much more efficient manner than before.

8 CONCLUSION

While WDL is a somewhat difficult language to learn upon initial use, it still allows for the production of efficient, scalable and reusable workflows. Throughout the translation of both the Whole Genome and 16s Workflows, I found it refreshing to have outputs configured automatically and inputs broadly stated. Giving rise to workflows that were not redundant in nature but quite modular, as users would be able to change and configure inputs as they pleased and such would be applied globally. However, due to the lack of information regarding its use WDL maybe difficult for novices who lack programming experience to use it effectively. The next steps for my project will be attempting to install and configure EPAM Cloud Pipelines, a Web based GUI management tool still being maintained, that can assist in the construction and management of WDL workflows. Such will allow for novices unfamiliar with WDL to be introduced to it in a more user friendly environment than the command line. Moreover, as WDL is still an evolving language I look forward to seeing its progress, new functionalities, and new tools added, as future versions are released, and will attempt to update my own workflow accordingly given the new capabilities provided.

9 ACKNOWLEDGEMENTS

My appreciation is extended to David Barbella for his assistance in helping me in articulating my proposal and research basis, as well as Charlie Peck's assistance in acquiring resources.

REFERENCES

[1] 2020. *Cromwell Subworkflows.* https://cromwell.readthedocs.io/en/stable/SubWorkflows/
[2] 2020. *The freedom of portable workflows.* https://terra.bio/the-freedom-of-portable-workflows/
[3] 2020. *MiniWDL.* https://miniwdl.readthedocs.io/en/latest/
[4] 2020. *OpenWDL.* https://openwdl.org/#three
[5] 2021. *Cromwell on AWS.* https://aws.amazon.com/government-education/cromwell-on-aws/#:~:text=Cromwell%20is%20a%20workflow%20management,%2C%20based%20in%20Cambridge%2C%20MA.&text=Cromwell%20is%20a%20workflow%20execution,tasks%20needed%20for%20genomics%20analysis.
[6] 2021. *Cromwell Workflow Manager and WDL Workflows.* https://sciwiki.fredhutch.org/compdemos/Cromwell/
[7] Xiaoling Chen and Jeffrey T Chang. 2017. Planning bioinformatics workflows using an expert system. *Bioinformatics* 33, 8 (2017), 1210–1215.
[8] Martín Garrido-Rodriguez, Daniel Lopez-Lopez, Francisco M Ortuno, María Peña-Chilet, Eduardo Muñoz, Marco A Calzado, and Joaquin Dopazo. 2021. A versatile workflow to integrate RNA-seq genomic and transcriptomic data into mechanistic models of signaling pathways. *PLoS computational biology* 17, 2 (2021), e1008748.
[9] Michael Jackson, Edward Wallace, and Kostas Kavoussanakis. 2020. Using rapid prototyping to choose a bioinformatics workflow management system. *bioRxiv* (2020).
[10] Samuel Lampa, Martin Dahlö, Jonathan Alvarsson, and Ola Spjuth. 2019. SciPipe: A workflow library for agile development of complex and dynamic bioinformatics pipelines. *GigaScience* 8, 5 (2019), giz044.
[11] Francesco Strozzi, Roel Janssen, Ricardo Wurmus, Michael R Crusoe, George Githinji, Paolo Di Tommaso, Dominique Belhachemi, Steffen Möller, Geert Smant, Joep de Ligt, et al. 2019. Scalable workflows and reproducible data analysis for genomics. In *Evolutionary Genomics*. Springer, 723–745.
[12] Laura Wratten, Andreas Wilm, and Jonathan Göke. 2021. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nature methods* 18, 10 (2021), 1161–1168.

A ONLINE RESOURCES

Git repositories of OpenWDL, Cromwell, MiniWDL and various open source workflow management tool implementations. Introductory series of getting started with OpenWDL.

[10] [8] [7] [9] [3] [11] [4]