

Scalable Pathogen Pipeline Platform (SP³)

Enabling Unified Genomic Data Analysis with Elastic Cloud Computing

Fan Yang-Turner^{1,2}, Denis Volk^{1,2}, Philip W. Fowler^{1,2}, Jeremy Swann^{1,3}, Matthew J. Bull⁴, Sarah Hoosdally¹, Thomas R. Connor⁴, Tim Peto^{1,2,3} and Derrick Crook^{1,2,3}

¹Nuffield Department of Clinical Medicine, University of Oxford, Oxford, United Kingdom

²NIHR Oxford Biomedical Research Centre, John Radcliffe Hospital, Oxford, United Kingdom

³NIHR Health Protection Research Unit in Healthcare Associated Infections and Antimicrobial Resistance at University of Oxford in partnership with Public Health England, Oxford, United Kingdom

⁴School of Biosciences, Cardiff University, Cardiff, United Kingdom

Abstract—Pathogen genomic data analysis can be extremely bespoke and diverse. This paper presents our plan and progress towards creating a Scalable Pathogen Pipeline Platform (SP³) providing an efficient and unified process of collecting, analysing and comparing genomic data analysis with the benefit of elastic cloud computing. SP³ enables container-centric bioinformatic workflows run on personal computers, High-performance computing (HPC) clusters and cloud platforms. We have deployed and tested SP³ on local HPC, Google Cloud Platform (GCP), Microsoft Azure and OpenStack Platforms. SP³ allows users to fetch genomic sequencing data from European Nucleotide Archive (ENA) and conduct analysis with open-source bioinformatic pipelines. We believe SP³ will promote common standards around pathogen genomic data quality, data processing and data analysis, helping answer the challenges of tools divergence and leveraging a pool of public genomic data repository and cloud resources.

Keywords—*pathogen genomic analysis; whole genome sequencing; elastic cloud computing; software-as-a-service*

I. INTRODUCTION

With emerging technologies such as container and cloud computing, the bioinformatic community has made great progress advancing genomic data analytics. For example, cloud-based infrastructure has been proposed and implemented to provide large dataset hosting and analysis solution [1][2]. Container-based (like Docker [3] or Singularity [4]) workflows can be deployed on cloud-based VMs, ready to perform analysis within a few minutes, highly simplifying deployment strategy [5][6][7]. For reproducible research, a wide range of tools for constructing bioinformatic workflows as well as containerizing bioinformatic tools are available [8][9]. Among the tools, Nextflow [10], a portable, scalable, and parallelizable domain-specific language, has been adopted by the bioinformatic community because it produces flexible, reproducible and extensible workflows. Nextflow-based bioinformatic pipelines have been developed and proved to be effective and efficient [11].

Implementation of fetching and data analysis in a cloud environment remains a challenge at an organization level. For example, although the consistent, portable execution environment provided within a container resolves issues caused by differences between cloud environments, there are no off-shelf solutions to enable users to run container-based pipelines

on different cloud providers. In the context of pathogen genomic data analysis, only preliminary work has been published on providing cloud-based solutions for large scale of data processing, analysis comparisons and reproducible research [12].

II. PATHOGEN GENOMIC DATA ANALYSIS

Pathogen genomic data analysis can be extremely bespoke and diverse. Many institutions and labs around the world have their own infrastructure and software suites to collect, process and analyze data. However, those existing solutions are difficult to use outside of the organization and the datasets for clinically validating software are also not standardized. This has hindered the research and clinical service utilizing the whole genome sequencing technology for pathogen identification and diagnosis.

We have designed and developed Scalable Pathogen Pipeline Platform (SP³), an open source web-based platform that enables container-centric bioinformatic workflows running on a diverse set of platforms, from single PCs to cloud infrastructure. It fetches data from a public repository like ENA [13] and configures bioinformatic workflows written with container-technology, like Docker or Singularity. It maximizes usability and minimizes software and infrastructure maintenance. It exhibits an always-on managed software as a service (SaaS) where the analysis can be achieved by a few clicks without having to maintain the underlying software or hardware. When deployed to commercial cloud platform, it benefits from elastic cloud computing, allowing users to pay only for data processing done, rather than a fixed maintenance cost of traditional clusters. As an open platform, SP³ aims to promote the reproducible, scalable and maintainable bioinformatic pipeline development. The platform is configurable and extendable through modularization, containerization and cloud-centric deployment. It utilizes the best regarded bioinformatic pipelines available in the pathogen research community and is establishing a growing user base of pathogen research, clinical diagnosis and public health communities.

III. PLATFORM FEATURES

To use SP³, a researcher can: (1) fetch the data from ENA to a cloud platform; (2) select one or more deployed pipelines for genome analysis; (3) Run the chosen pipeline and monitor

the progress; (4) Download the analysis to local storage. SP³ provides a web-based user interface operating on elastic cloud computing, offering maximum portability among execution environment. It has following features:

A. Execution on Elastic Cloud Computing

SP³ scales the resources on demand. When there is no task, the resources will be deallocated and when there are requests for running one or more bioinformatic pipelines, the platform will allocate resources based on the pipeline CPU and memory requirements. SP³ adopts Slurm elastic computing [14] as a unified execution layer of abstraction for users over cloud-based infrastructure, eliminating the need for provider-specific cluster management. Slurm has the ability to support a cluster that grows and shrinks on demand. This ability can be realized on Google Cloud Platform (GCP) and Amazon Elastic Computing Cloud (Amazon EC2).

B. Plug and Play via Declarative Configuration

SP³ promotes portability and scalability of bioinformatic pipeline development. It supports any bioinformatic pipeline developed using Nextflow paired with container technology, such as Docker or Singularity. It allows users to configure the bioinformatic pipeline using declarative configuration via a YAML formatted file. The configuration file describes how the pipelines should run, such as the input and output parameters and the scripts. A YAML extension is used to enable users to include the specific pipeline configuration in the main configuration file. The platform parses the configuration file and provides web-based user interface to run the pipelines. At runtime, further checks are made, for example, the pipeline version is obtained from the pipeline's git repository.

C. Connect to Public Repository

Public repositories, like ENA [13], NCBI [15] and DDBJ [16] are used to store whole genome sequences and are synchronized on a daily basis. SP³ has built-in support to fetch data from ENA to the execution platform. After fetch, the user can choose one of the pipelines to run against the dataset. The fetched data persists for a period of time, allowing users to run different pipelines, and then deleted after all the analyses have been performed. The fetch API is modular and extendable, which allow developers to fetch data from other data sources.

D. Composing Bioinformatic pipelines

One of SP³ advances is to compose bioinformatic pipelines via declarative configuration. Pipelines can be composed by providing additional configuration such as the relative subdirectory of its output, which will be used as the input for the next pipeline. Adding a composite pipeline can then be accomplished by simply creating a pipeline configuration file and listing the pipelines to be run in sequence.

The web interface for creating a new run collects and presents all the pipeline parameters needed in one form, so the composite pipeline is presented and acts as one pipeline.

E. Tracking and Monitoring

SP³ tracks and monitors the running bioinformatic pipeline. It provides instant output as the pipeline executes. It not only

displays Nextflow reports and timelines but also offers detailed logs of individual pipeline processes, showing exactly which command was run and what the output was to the web users, helping users debug issues. The platform presents the run information grouped by sample name and the sample's corresponding processes.

F. Download the Output

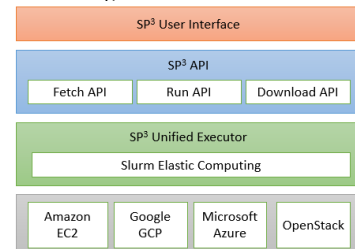
Upon the completion of each step of the pipeline, users can download the output via a web user interface. Files are served with the Nginx web server [17], configured to authenticate against a Download API, allowing users to specify output data access restrictions (open access, basic web authentication, LDAP authentication). To help researchers download files in bulk, the web user interface provides a command line that can run and with filter options to download only files that of interests.

IV. PLATFORM IMPLEMENTATION

A. Architecture

The architecture of SP³ consists of four layers (Fig. 1). The UI layer, written on Python and Flask, provides a modern, performant and intuitive interface for running bioinformatic pipelines. The API layer consists of three sub-projects: the Fetch API, responsible for downloading data from diverse bioinformatic data sources; the Run API, which supervises the Nextflow processes; and the Download API to download pipeline output files with optional authentication. The unified executor layer is in the form of Slurm configured for elastic computing, which creates and deallocates nodes on demand. The cloud platform layer, which Slurm elastic computing can be realized on, maybe from any provider, for example: GCP, AWS, Azure and OpenStack.

Fig. 1. SP³ Architecture



B. User Interfaces

In this section, we present SP³ user interfaces to showcase a typical user journey to run pathogen pipelines. A user could start with fetching data (Fig. 2). The platform will show the datasets details with full meta-data provided by ENA. The user can start a new fetch or use the downloaded data for analysis. The platform is equipped with a few pipelines from pathogen research communities. The user can select one or more for their analysis. When starting a new run, a user will customize a pre-populated form with the chosen run parameters (Fig. 3). After a run is started, the user can look at the progress of the analysis steps on the run details page (Fig. 4), and select one of the options for details, such as log, output, reporting etc. For each sample running through, an instant output of each step is available to view in the logs while processing.

Fig. 2. Datasets

Datasets				
New Fetch				
Submit time	Kind	Name	Progress	Status
2019-01-31 15:42:04	ena1	PRJNA302362	2 / 2	success
2019-01-24 14:59:13	ena1	PRJNA324238	20 / 20	success
2019-01-21 10:59:15	ena1	PRJNA268101	2 / 2	success
2019-01-30 22:10:45	ena1	PRJEB9025	20 / 20	success
2019-01-21 18:03:20	ena1	ERR841881	2 / 2	success
2019-01-17 16:34:47	ena1	PRJEB25968	66 / 66	success
2019-01-21 18:04:54	ena1	PRJEB9025	20 / 20	success
2019-01-31 12:15:27	ena1	PRJEB25968	2 / 2	success
2019-01-17 16:35:44	ena1	PRJEB25968	28 / 28	success
2019-01-21 12:50:57	local1	/mnt/disk1/inputs/compassnext-bams	20 / 20	success
2019-01-27 21:01:14	ena1	PRJNA315363	6 / 6	success
2019-01-28 16:27:51	local1	/mnt/disk1/inputs/sim_collection	100 / 100	success

Fig. 3. New Run User Input

Create a new run for bug-flow

Run Name

Execution context

Read Directory

Input file pattern (nextflow syntax, e.g.: *.fastq.gz or *.bam)

Reference file

[Submit](#)

Fig. 4. Run Details

Run details	
Bug Flow / bug-flow-20190128_143034	
View log View files View report View timeline Fetch Output Repeat Run	
Sample	Process
NC_000962_2.fasta	indexReference
ERR2509675	rawFastQC tbbDuk cleanFastQC bwa removeDuplicates mpileup
ERR2509674	snpCall filterSnps consensusFa spades
ERR2509673	rawFastQC tbbDuk cleanFastQC bwa removeDuplicates mpileup
	snpCall filterSnps consensusFa spades

V. FUTURE PLAN

SP³ enables unified pathogen genomics data analysis in research institutes and public health organizations without the need for extensive local software development expertise. We are in the process of collecting users' feedback from the pathogen genomics research community and iteratively improving the features of the platform. We believe SP³ solution will facilitate standards around how the pathogen genomic data are being processed, compared and validated and to ensure that results generated in one laboratory are comparable to those generated elsewhere. The current work of SP³ has enabled us to work towards an end-to-end solution of pathogen genomics widely adopted by clinical settings. To achieve that, SP³ will provide further technical facilities for integrating the applications of identifying the organism causing certain infections, predicting how it might respond to treatment and

how that organism may be related to other organisms of interest.

ACKNOWLEDGEMENTS

This research was funded by the National Institute for Health Research (NIHR) Oxford Biomedical Research Centre (BRC) and the National Institute for Health Research Health Protection Research Unit (NIHR HPRU) in Healthcare Associated Infections and Antimicrobial Resistance at University of Oxford in partnership with Public Health England (PHE). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR, the Department of Health or Public Health England. Tim Peto is a NIHR Senior Investigator. This work made use of computational resources provided by MRC CLIMB, which also funds TRC and MJB (grant reference MR/L015080/1).

REFERENCES

- [1] Afgan E. et al., "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update," *Nucleic Acids Res.*, vol. 44, no. W1, pp. W3–W10, 2016.
- [2] Weinstein J. N. et al. (2013) The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* 45, 1113–1120
- [3] Merkel, D.: Docker: lightweight Linux containers for consistent development and deployment. *Linux J.* 2014 (239), 2 (2014)
- [4] Kurtzer GM, Sochat V, Bauer MW (2017) Singularity: Scientific containers for mobility of compute. *PLoS ONE* 12(5): e0177459.
- [5] O'Connor BD, Yuen D, Chung V et al. The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows. *F1000Research* 2017, 6:52
- [6] Belmann P, Dröge J, Bremges A, McHardy AC, Sczyrba A, Barton MD. Bioboxes: standardised containers for interchangeable bioinformatics software. *Gigascience.* 2015;4:47.
- [7] Di Tommaso P, Palumbo E, Chatzou M, Prieto P, Heuer ML, Notredame C. The impact of Docker containers on the performance of genomic pipelines. *PeerJ.* 2015;3:e1273.
- [8] Leipzig J. A review of bioinformatic pipeline frameworks. *Brief Bioinform.* 2016.
- [9] Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics.* 2012;28:2520–2.
- [10] Di Tommaso P. et al. (2017) Nextflow enables reproducible computational workflows. *Nature Biotechnology*, 35(4), 316–319.
- [11] Zhao Q. et al. LncPipe: A Nextflow-based pipeline for identification and analysis of long non-coding RNAs from RNA-Seq data. *J Genet Genomics.* 2018 Jul 20; 45(7): 399–401
- [12] Yang-Turner F. et al., "An Open-Source Azure Solution for Scalable Genomics Workflows," *2018 IEEE World Congress on Services (SERVICES)*, San Francisco, CA, 2018, pp. 39–40
- [13] Stoesser, G., M. A. Moseley, J. Sleep, M. McGowran, M. Garcia-Pastor, and P. Sterk. 1998 . The EMBL nucleotide sequence database. *Nucleic Acids Res.* 26:8–15.
- [14] Slurm Elastic Computing (Cloud Bursting): https://slurm.schedmd.com/elastic_computing.html (accessed on May 2019)
- [15] Geer L.Y. et al., The NCBI BioSystems database. *Nucleic Acids Res.* 2010 Jan; 38 (Database issue): D 492–6.
- [16] Tateno Y. et al. (2002). "DNA Data Bank of Japan (DDBJ) for genome scale research in life science". *Nucleic Acids Res.* 30 (1): 27–30.
- [17] Reese, W. Nginx: The high-performance web server and reverse proxy. *Linux J.* 2008, 9, 1–4