PharmaSUG 2023 - Paper SD-207

Acceleration and automation of genomic data analysis to meet corporate compliance standards using advanced cloud components.

Dr. Gopal Joshi, Mr. Satyoki Chatterjee, Mr. Pankaj Choudhary, Sanjay Koshatwar, Shekhar Seera Circulant solutions Inc., Pleasanton, CA 94588, USA

ABSTRACT

Recent advancements in High-throughput next-generation sequencing (NGS) technologies grew exponentially in genomic research revolutionizing biological data analysis, and enhancing the study of complex biological systems at an unprecedented scale. The technological limitations of the NGS system are the deluge of genomic data produced. It's difficult for a single workstation to execute sequential methods and produce results quickly. Efficacy decreases significantly with human interference and to mitigate them, we developed an in-house pipeline, with the help of AWS services and tools like snakemake, kallisto, etc., for automating RNA-seq data analysis. It's efficient, scalable, reproducible, version-controlled, transparent, and cost-effective for large volumes of data. In this study, we have reviewed the RNA-sequencing technique using AWS to analyze gene expression at the transcriptional level. The systematic approach allows CROs to transfer raw data using an SFTP server, followed by an automated transfer to Simple Storage Service (S3) and preceded by data quality validation. Helper scripts then transfer data from S3 to Elastic File System (EFS), launch the Fastq processing pipeline, clone a GitHub repo of the corresponding project, and leverages AWS Batch to spin up a dynamic Elastic Compute Cloud (EC2) instance as desired. After successful execution, outputs are available in EFS, and actual data analysis is performed using RStudio Workbench ending with automated results archival in S3.

INTRODUCTION

In recent years, there have been advancements in High-throughput next-generation sequencing (NGS) technologies which have made exponential growth in genomic research and have revolutionized biological data analysis, allowing researchers to study complex biological systems at an unprecedented scale. (Zhao et al., 2017). Biological sciences have been transformed with the combination of NGS technology in biological research. It has evolved biological data science into big data science. With the introduction of NGS technology, biological research has evolved into a big data field where computations supplement the short read and high error rate of experimental research with high-depth coverage sequencing data. The capability and resolution of many biological, pharmaceutical, diagnostic, and medical applications, such as genome-wide sequencing, quantitative transcriptome analysis (RNA-seq), identification of protein binding sites (ChIP-seq), genome-wide methylation profiling, and the assembly of large genome or transcriptome data, have recently been significantly impacted by the rapid development of new sequencing technologies. RNA-seq is one of the widely used sequencing techniques used to map and identify and understand their regulation and roles across species. (Emrich et al., 2007; Lister et al., 2008; Zhang & Jonassen, 2020). Functional genomics has marked a significant emphasis on gene expression, a process that has been extensively studied. The evolution of genetic information from the genomic DNA template to useful protein products is referred to as gene expression(Griffith et al., 2015). For examining relative transcript abundance and diversity, RNA sequencing (RNA-seq) has established itself as a standard gene expression analysis. Gene expression datasets are readily available in databases like GEO (Barrett et al., 2011) and ArrayExpress(Athar et al., 2019) as a result of numerous research that has been conducted utilizing RNA-Seq. RNA-seq produces short reads from the fragments. Millions of relatively small reads are generated by RNA-Seq experiments from the ends of cDNAs created from RNA sample fragments. The reads can be utilized in a number of applications such as transcriptome analyses, including transcript quantification(Bohnert & Rätsch, 2010)(Trapnell et al., 2010), differential expression testing(Anders & Huber, 2010)(Robinson et al., 2010), reference-based gene annotation(Trapnell et al., 2010) (Guttman et al., 2010), and de novo transcript assembly(Robertson et al., 2010)(Grabherr et al., 2011)(Li & Dewey, 2011). When RNA-Seq is employed in a study, the sequencing reads go through several processing and analysis procedures. The phases are frequently arranged into a workflow that can be partially or entirely automated. The procedures are as follows: quality control (QC) and trimming, mapping of reads to a reference genome (or transcriptome), quantification at the gene (or transcript) level, statistical analysis of expression statistics to report genes (or transcripts) that are differentially expressed between two preset groups of samples(Stark et al., 2019). Genome alignment requires extensive time and computation which has been overcome with the help of pseudo-alignment using different tools. When compared to normal alignment it has been demonstrated that pseudo-alignment improves gene expression estimates while also being more computationally efficient(Zhang & Jonassen, 2020).

A gradual evolution of biological sciences has created significant progression in NGS technologies such as Whole genome sequencing, RNA sequencing, whole exome sequencing, etc. There is the generation of high-volume data which requires powerful computing resources to analyze and process the huge biological data. It is getting increasingly difficult for a single workstation to execute sequential methods and produce results in a reasonable amount of time. To deal with computationally demanding tasks, cloud computing is a possible approach. To automate and streamline the NGS-based technologies such as the RNA-seq data analysis process, one can use cloud computing services such as those offered by Amazon Web Services (AWS), Google Cloud Platform (GCP), and workflow management tools like Snakemake. AWS offers a range of services including Elastic Compute Cloud (EC2) for running computational tasks, Simple Storage Service (S3) for storing and accessing data, and Batch for automating and scaling up analysis jobs. Snakemake, an open-source workflow management tool, allows to define and execute complex workflows in a reproducible and scalable manner. By combining AWS services with Snakemake, the creation of highly automated and efficient RNA-seq data analysis pipelines is possible. With the introduction of NGS technology, biological research has evolved into a big data science where computational calculations supplement the short read and high error rate of experimental research with high-depth coverage sequencing data (Kwon et al., 2015). Snakemake is a robust workflow engine that makes workflow management simple. It breaks down the entire workflow into rules with each rule carrying out a particular step (Zhang & Jonassen, 2020). Each rule has an input for specifying the input files and output for specifying the output files and a shell for specifying the command used to generate the output according to the input. The execution of a Snakemake pipeline is accomplished through the specification of a single target file name. To produce the target output Snakemake determines the flow based on its rules, the file name, and the application of the wildcards concept(Mohsen et al., 2022). Snakemake workflows have a wide range of applications and can be utilized to automate bioinformatics tasks efficiently.

In this paper we illustrate the concepts and principles of cloud computing and describe an instance of RNA-seq data analysis on cloud services computing with integration to snakemake. In addition, we surveyed challenges caused by data generated through RNA-seq and have proposed a promising solution to overcome the limitations related to storage, manual error, time, and performance by integrating cloud computing. The introduction of cloud computing in large-scale RNA-seq data analysis has helped users perform their tasks efficiently without expending time and resources. It also offers them an alternative to the traditional methods of analysis.

BACKGROUND

"Cloud computing" refers to the on-demand distribution of IT resources and applications through the Internet with pay-as-you-go billing. Cloud computing is a concept for providing ubiquitous, on-demand access to a shared pool of customizable computing resources (e.g., networks, servers, storage, applications, and services) that can be deployed and released quickly with minimal administration effort. Cloud computing eliminates the need for expensive hardware and management. Cloud services, such as AWS, provide network-connected hardware as well as the optimal computational specifications required to accomplish the task(Mrozek, 2020). With AWS, you can instantly access as many resources as you require and only pay for what you request and own. Networks, servers, storage, applications, and services are examples of computing resources. Some characteristics of the cloud computing model according to the National Institute of Standards and Technology (NIST) (Mell & Grance, 2011) are as follows:

1. On-demand self-service: The user can use the resources according to their need without any assistance from cloud provider staff.
2. Broad network access: Cloud services provides broad network connectivity, allowing users to accomplish the task from any network including tablets, workstation or laptops etc.

3. Resource pooling: the services required to perform the task can be pooled and utilised by number of individuals according to their need.
4. Rapid elasticity: users can allocate the resources as per their requirements and can dynamically scale up and down with time.
5. Measured service: the services utilised are routinely measured, monitored, and regulated, and the user may obtain a cost estimate and a report on the resources utilized.

In other words, Cloud services provide rapid scaling out or scaling up of RNA-seq analysis, allocating resources as needed for speeding up analyses at a remote data centre with minimum effort, over the network, and optimizing their allocation to mitigate usage costs(Zhao et al., 2017). This eliminates the requirement to purchase large computer clusters that may remain inactive for certain times, as well as the expenditures associated with the purchase and maintenance of the hardware.

CLOUD COMPUTING RESOURCES FOR RNA-SEQ DATA ANALYSIS

Cloud computing services are constantly growing and introducing new solutions for challenging purposes. Cloud platforms such as AWS, Azure, and GCP provide a plethora of resources. Different resources are utilized by users according to their computational requirements for performing experiments. Cloud computing is classified into two types of cloud and cloud technologies. The cloud provides a big pool of easily manageable and customizable resources that are scalable to allow for optimal utilization. Virtualization is a key foundation of cloud technologies, a single physical system that can host several virtual machines (VMs). A virtual machine (VM) is a software application that can simulate a real computer environment by installing a single digital picture of resources, often known as a whole system snapshot. Furthermore, a VM image can be completely replicated, including the operating system (OS) and any related applications(Zhao et al., 2017). While using AWS the user identifies and manages his virtual machines within a virtual network consisting of a subnet, a set of virtual network interfaces, and a pool of public and private IP addresses. Among the variety of resources that can be deployed from the Cloud, the following can be employed in common scenarios involving RNA-seq data analysis:

- Storage space
- Compute resources
- Web services

MATERIALS AND METHODS

TECHNOLOGY STACK USED

During our study various resources were deployed which are mentioned below:

AWS SFTP: The SFTP is also known as SSH File Transfer Protocol (Secure File Transfer Protocol), which is a network protocol that allows users to access, transfer, and manage files via any secure data stream. Vendors upload biological data into a predefined folder structure of AWS SFTP, followed by a cron job transfer to Amazon S3.

Amazon S3: Amazon S3 is an object storage service that provides industry-leading scalability, data availability, security, and performance. Customers of all sizes and sectors may use Amazon S3 to store and safeguard any amount of data for a variety of use cases, including data lakes, backup and restore, archiving, and big data analytics. Moreover, Amazon S3 has management options that allow you to optimize, organize, and configure data access to suit your unique business, organizational, and compliance needs(Palankar et al., n.d.). In this study, S3 is used to store large volumes of biological data such as fastq and metadata.

Amazon EFS: Amazon Elastic File System (Amazon EFS) enables serverless, completely elastic file storage, enabling you to exchange file data without having to provide or manage storage capacity and performance. Amazon EFS is designed to expand to petabytes on demand without affecting applications, growing, and shrinking dynamically as files are added and removed. Because Amazon EFS features a simple web services interface, you can quickly and easily establish and configure file systems. The service maintains all your file storage infrastructure, saving you the trouble of deploying, patching, and maintaining complicated file system settings. In this study, EFS is mounted on EC2 to store, access, and manage files efficiently.

Amazon EC2: Amazon Elastic Compute Cloud (Amazon EC2) is a computational resource that is scalable on the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates the need to invest in infrastructure upfront, allowing you to create and deploy services more readily. You can deploy as many virtual servers as you need, set security and networking, and manage storage using Amazon EC2. It allows you to scale up or down in response to variations in demand or surges in popularity, minimizing the need to assess traffic. In this study an instance is created on EC2 which acts like a virtual machine to perform operations with cloud features while running the pipeline.

AMI: An instance can be launched using an Amazon Machine Image (AMI), which is a supported and maintained image offered by AWS. When you launch an instance, you must provide an AMI. When you need numerous instances with the same configuration, you may launch them all from a single AMI. When you need instances with specifications, you may start them using multiple AMIs. It is used to expand the disc's size so that processing huge volumes of data is quicker.

An AMI includes the following:

1. A template for the instance's root volume, or one or more Amazon Elastic Block Store (Amazon EBS) snapshots, instance-store-backed AMIs.
2. Provides launch permissions which AWS accounts can utilize the AMI to launch instances.

AWS CLI: The AWS Command Line Interface (AWS CLI) is a free and open-source tool that allows you to interface with AWS services using command-line shell commands. The AWS CLI allows you to start executing commands that provide functionality equivalent to that given by the browser-based AWS Management Console from the command prompt in your terminal program with minimal configuration. It is used to interact with AWS using the config and credential files which includes security credentials, the default Output format, and the default AWS Region.

Docker: Docker is an open-source platform for executing applications and simplifying the development and distribution process. Docker grants the ability to automate the deployment of applications into Containers. Docker offers an extra layer of deployment engine on top of the virtualized and performed programs in a Container environment. It is meant to provide a rapid and lightweight environment in which code can be executed expediently, as well as an additional facility for the proficient work process to take the code from the computer for testing before production. In this study, docker is used to create an image with necessary libraries using a docker file and shell script which is later pushed to Amazon Elastic Container Registry (ECR).

Python: Python version 3.8 is used for creating virtual environments and for writing scripts required for running the pipeline.

RHEL: Red Hat Enterprise Linux (RHEL) version 8 for running the pipeline. It is a business-oriented Linux operating system (OS) developed by Red Hat. RHEL facilitates users with a stable, dependable base across environments. It is incorporated with all the capabilities required to offer application services and workloads promptly. RHEL, like all Linux distributions, is built on a free and open-source platform.

Amazon ECR**:** Amazon Elastic Container Registry (Amazon ECR) is a secure, scalable, and credible container image registry service provided by AWS. It enables private repositories with AWS IAM resource-based permissions. This is implemented so that only certain users or Amazon EC2 instances can access your container repositories and images. Using any CLI one can push, pull and manage docker images. In the present work, AWS ECR is implemented to push docker images into the container, which assists in initiating the script that would facilitate the pipeline's execution.

Amazon ECS: Amazon Elastic Container Solution (Amazon ECS) is a container management service that is extremely scalable and instant. It can be used to start, stop, and manage containers on a cluster. Containers in Amazon ECS are described in a task definition, which is used to perform a particular job within a service.

Amazon batch: AWS Batch enables the execution of batch computing workloads on the AWS Cloud. Batch computing is a common approach for developers, scientists, and engineers to gain access to enormous amounts of computing resources. AWS Batch removes the undifferentiated heavy lifting of establishing and managing the requisite infrastructure which is used in traditional batch computing tools. To overcome capacity restrictions, decrease compute costs, and provide rapid results, this service can efficiently deploy resources in response to submitted jobs.

AWS batch is configured using the following attributes:

1. Job Definition: AWS Batch job definitions define the steps involved in running a task. Even though each job must refer to a job specification, many of the parameters listed here can be changed at runtime.
   Some of the attributes specified in a job definition include:
   - Docker image to use with the container in your job.
   - The Required number of vCPUs and memory to use with the container.
   - Data volumes that should be used with the container.
   - IAM role to be used by the job for AWS permissions.

2. Compute Environment: Various computing environments are assigned to various job queues. Containerized batch jobs are executed in computing environments using Amazon ECS instances. One or more job queues may also be assigned to a particular computing environment. The scheduler uses the order of the related computing environments inside a job queue to decide which environment will process a job that is ready to be executed.
3. Job Queue: Jobs are submitted to a job queue, where they reside to be scheduled to run in a computing environment. Multiple job queues can exist in an AWS account. The scheduler uses the priority assigned to each job queue to choose which ones should be assessed for execution first. The name, state, priority, and order of the computing environment are the four parameters of a task queue.

Amazon cloud watch: CloudWatch is a management and monitoring service for AWS, providing real time action insights. All performance and operational data can be tracked and accessed in the form of logs and metrics on one single platform rather than server or database. It provides up to one-second access to metrics and log data. CloudWatch enables you to monitor your whole stack (applications, infrastructure, network, and services) and use alarms, logs, and events data to automate actions and reduce mean time to resolution (MTTR).

RStudio Workbench: RStudio workbench version 2022.07. 2+576.pro12 is used for the analysis of data. Enterprise-level integrated development environment for data is known as RStudio Workbench, scientists who need to build, collaborate, and scale in R and Python utilize it efficiently. On workbench, Professionals operate from a centralised server in their preferred language and with the computer resources that they require. In the present work we have used RStudio workbench for the analysis of the results which are generated after the processing of the RNA-seq pipeline. Later, the results are visualised using plots and graphs. It uses R code or R markdown, and R env for package management with mounted EFS.

GitHub: Git version 2.31.1 is used in the pipeline. It is a platform for hosting code that allows for version control and collaboration. It allows you and others to collaborate on projects from anywhere.

FileZilla: FileZilla is a free FTP client that offers customers a convenient, multi-interfaced solution for file transfers via FTP. It allows users to upload, download, edit, and delete files all at single platform.

BIOINFORMATICS TOOLS:

Cutadapt: Cutadapt version 3.7 is used in the pipeline. It searches and eliminates adapter sequences, primers, poly-A tails, and other forms of irrelevant sequences from raw RNA-seq data. It aids in trimming by locating adapter or primer sequences in an error-tolerant way. It can also modify, and filter reads in a variety of ways. It also de-multiplexes the input data without discarding any adapter sequences(Martin, 2011).

Kallisto: Kallisto version 0.46.2 is used in the pipeline. It is a software for measuring transcript abundances from bulk and RNA-Seq data, or more broadly, target sequence abundances using high-throughput sequencing reads. It is based on the new principle of pseudoalignment, which allows the quick determination of read compatibility with targets without the necessity for alignment(Bray et al., 2016).

Snakemake: Snakemake version 6.5.3 is used as workflow management tool. The Snakemake workflow management system is a tool for creating robust and reproducible data analyses. Workflows are specified using a Python-based human-readable language. They can be scaled across server, cluster, grid, and cloud environments without requiring any changes to the workflow architecture. It can include a description of the software that must be installed, which will be automatically distributed to any executable environment (Mölder et al., 2021).

WORKFLOW:

In this study, to automate the processing of RNA-seq data and to calculate the abundance of transcripts (Figure 1), we deployed AWS services. Initially, python version 3.8 is installed for the creation of a python virtual environment on an AWS EC2 instance. Later, the Python virtual environment is activated, and it is scheduled to be launched every time the server is initialized. We also implemented AMI, a service offered by ec2, to expand the disc capacity so that we can effectively process huge volumes of data. Users can use AMIs developed by AWS or can freshly create and rapidly launch new instances with all the necessary resources. Tools and libraries such as Docker, AWS CLI etc. required for running the pipeline are installed. Eventually, all tools are configured according to the pipeline. In our systematic approach, Vendors transfer biological raw data to an AWS SFTP server via FileZilla, into a predefined folder structure followed by a cron job transfer to S3 accompanied by DQM validation (md5 checks). Data from S3 is accessed and transferred to Elastic file System (EFS) using an in-house developed suite of Python-based command line helper scripts. For running the pipeline, Docker image is created using docker file in which a shell script is copied which contains the code to build the directories and pull the repositories from GitHub for a particular study. This workflow is customized to accept multiple sequence file formats. Before pushing the docker image into AWS ECR, a repository needs to be created. It is crucial to configure AWS Batch with job definition, job queue, and compute environment before running the pipeline as compute environment will be dynamically scaled up or down according to the pipeline. Multiple jobs can be executed concurrently at the same time for various kinds of studies. An in-house developed Python script is employed for examining job specifications, creating, and submitting jobs to AWS Batch. This python script leverages AWS Batch to spin up a dynamic Elastic Compute Cloud (EC2) instance according to user-specified performance needs. As the batch job starts running, the fastq processing pipeline that is the shell script copied into the docker image is launched which creates work directories, clones a GitHub repo of the corresponding project containing snakemake pipeline and after, successful completion of the pipeline, the output results are stored back in EFS, so that the data analysis is performed using RStudio Workbench.

When the pipeline reaches the instance where the Snakefile is initialized, two bioinformatic tools Cutadapt and Kallisto start the processing of fastq files. Cutadapt searches and eliminates adapter sequences, primers, poly-A tails whereas Kallisto measure transcript abundances from bulk RNA-Seq data. The final abundance of transcript result marks the successful completion of the pipeline followed by an email notifying the user of the completion and a cloud watch link that allows the user to see the event logs. Moreover, if any error occurs during the process a failure email is triggered notifying the user regarding the same. The results obtained after the processing of fastq files are further analyzed on RStudio Workbench. The data analysis results such as plots, csv, graphs etc are moved back to S3 for long term archival using CLI helper scripts.

To benchmark the local workstation and cloud services, we ran Rna-seq analysis on the local workstation with instance type EC2 as t2.2xlarge with 8 vCPUs, 32 GiB of memory and storage of 105 GB, on cloud services (AWS EC2; https://aws.amazon.com/ec2/) (Table 1)

For the benchmark, we used the RNA sequencing data of human of size 4.2 GB.

The Rna seq data is composed two paired-end reads each of around 2 GB. The instance type of Amazon EC2 was t3.large with 2 vCPUs, 8.0 GiB of memory and up to 5 Gibps of bandwidth storage:1000 GB

Table 1. Specifications and Benchmark Results of the Local and Cloud Computing Services

Provider	**CPUs**	**RAM(Gbytes)**	**Storage (Gbytes)**	**OS**	**Cost ($)/h**	**Times(hh:mm:ss)total process**
Local	8	32	105	Red Hat Enterprise Linux Server release 8.6	NA	02:04:00
AWS EC2 (t3.large)	2	8	1000	Red Hat Enterprise Linux Server release 8.7	$0.0835	00:22:17

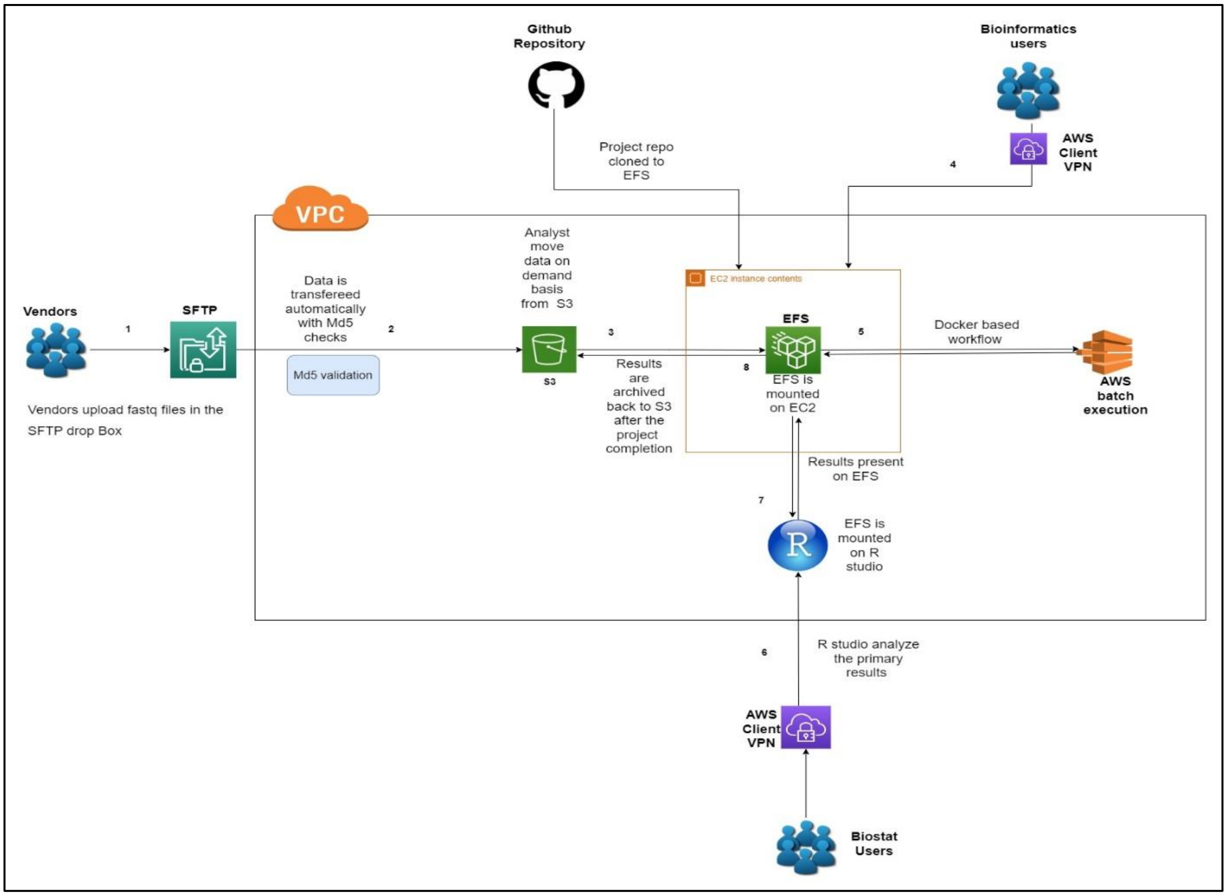
Github Repository
Bioinformatics users
Project repo cloned to EFS
4
AWS Client VPN
VPC
Analyst move data on demand basis from S3
EC2 instance contents
Data is transfereed automatically with Md5 checks
Vendors
1
SFTP
2
3
EFS
5
Docker based workflow
Md5 validation
S3
Results are archived back to S3 after the project completion
8
EFS is mounted on EC2
AWS batch execution
Vendors upload fastq files in the SFTP drop Box
Results present on EFS
7
EFS is mounted on R studio
6
R studio analyze the primary results
AWS Client VPN
Biostat Users

Figure 1. Architecture Diagram of RNA-Seq Workflow Performed in the Pipeline.

RESULTS AND DISCUSSION

Traditional methods of NGS face several difficulties in terms of storage, transportation, analysis, and cost efficiency. NGS generates a vast amount of data that is challenging to process using a single workstation in a reasonable amount of time. RNA-seq, for example, is traditionally performed with human intervention, which slows down the process and increases the risk of manual error. To address these challenges, the development of highly automated pipelines for data processing is crucial. A comprehensive RNA-Seq analysis pipeline (RAP) has been developed using cloud services like AWS to address the challenges of NGS. Cloud computing offers solutions to the difficulties related to storage, transportation, analysis, and cost efficiency. The growth in the volume and complexity of NGS data requires cluster or high-performance computing systems for analysis, but the cost of infrastructure and maintenance may be prohibitive for smaller institutions or laboratories and even for large institutions and pharmaceutical corporations. Cloud computing reduces infrastructure costs both initially and ongoing. The generation of large amounts of data from NGS platforms poses difficulties in processing, especially as the data size grows. AWS offers an unlimited storage capacity, making data storage less of a challenge. Conventional sequencing techniques, such as RNA-seq, are often slowed down by manual intervention and the risk of human error. By automating the sequencing process through cloud computing, the need for manual intervention is reduced, minimizing errors. When conducting manual Sequencing analysis, real-time monitoring of the process is not possible, but with AWS cloud services, it is possible to follow the process through live logs in CloudWatch. In traditional methods, the analysis setup must be established before running the analysis. In contrast, cloud computing offers instances that can be dynamically created based on data needs. Custom pipelines can be designed specifically for assays like RNA-seq but can be modified to accommodate other assays as well and support a variety of fasta file formats. NGS pipelines are usually unstructured, but with the help of tools like Snakemake and Nextflow, they can be organized and streamlined when using AWS cloud computing. The integration of Snakemake results in a highly modular workflow. This allows advanced users to easily extract or expand parts of the workflow based on their specific research needs. They can also replace the tools used in the RNA-seq workflow with alternative tools to explore new pipelines for analyzing various forms of Sequencing data. AWS enables the efficient execution of multiple jobs at the same time.

FUTURE PROSPECTS

The advent of next-generation sequencing has enabled the concept of a single universal test to become a reality, with clinical and public health laboratories as well as researchers increasingly adopting this approach. In the past, researchers faced significant difficulties due to the expensive nature of sequencing and the computational challenges that came with it. Nevertheless, with the considerable reduction in sequencing costs and the accessibility of cloud computing, this method has become more attractive to researchers who can now easily integrate sequencing into their research plans while overcoming obstacles related to processing power and storage capacity. Today, next-generation sequencing platforms can generate vast amounts of data rapidly, allowing for the exploration of diverse biological inquiries. Such data includes gene function and regulation, the diagnosis and treatment of diseases, and omics profiling of individual patients to enable precision medicine. The enormous amount of data being generated presents several challenges in terms of storage, transportation, analysis, and cost. It has become increasingly difficult to process this data on a single workstation. As a result, cloud computing offers a promising solution for researchers dealing with computationally demanding problems, as it provides the means to overcome limitations in processing power and storage capacity. Regarding bioinformatics, cloud computing is still in its early stages and not restricted to the analysis of NGS data. The development of more powerful and user-friendly cloud platforms and programming models is being pursued to address complex scientific issues. Ultimately, scientists from all disciplines stand to benefit from the increased computational power available in this field.

REFERENCES

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*(10), R106. https://doi.org/10.1186/GB-2010-11-10-R106

Athar, A., Füllgrabe, A., George, N., Iqbal, H., Huerta, L., Ali, A., Snow, C., Fonseca, N. A., Petryszak, R., Papatheodorou, I., Sarkans, U., & Brazma, A. (2019). ArrayExpress update – from bulk to single-cell expression data. *Nucleic Acids Research*, *47*(D1), D711–D715. https://doi.org/10.1093/NAR/GKY964

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Muertter, R. N., Holko, M., Ayanbule, O., Yefanov, A., & Soboleva, A. (2011). NCBI GEO: archive for functional genomics data sets--10 years on. *Nucleic Acids Research*, *39*(Database issue). https://doi.org/10.1093/NAR/GKQ1184

Bohnert, R., & Rätsch, G. (2010). rQuant.web: a tool for RNA-Seq-based transcript quantitation. *Nucleic Acids Research*, *38*(Web Server issue). https://doi.org/10.1093/NAR/GKQ448

Bray, N. L., Pimentel, H., Melsted, P., & Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, *34*(5), 525–527. https://doi.org/10.1038/NBT.3519

Emrich, S. J., Barbazuk, W. B., Li, L., & Schnable, P. S. (2007). Gene discovery and annotation using LCM-454 transcriptome sequencing. *Genome Research*, *17*(1), 69–73. https://doi.org/10.1101/GR.5145806

Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B. W., Nusbaum, C., Lindblad-Toh, K., … Regev, A. (2011). Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nature Biotechnology*, *29*(7), 644. https://doi.org/10.1038/NBT.1883

Griffith, M., Walker, J. R., Spies, N. C., Ainscough, B. J., & Griffith, O. L. (2015). Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud. *PLOS Computational Biology*, *11*(8), e1004393. https://doi.org/10.1371/JOURNAL.PCBI.1004393

Guttman, M., Garber, M., Levin, J. Z., Donaghey, J., Robinson, J., Adiconis, X., Fan, L., Koziol, M. J., Gnirke, A., Nusbaum, C., Rinn, J. L., Lander, E. S., & Regev, A. (2010). Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nature Biotechnology*, *28*(5), 503–510. https://doi.org/10.1038/NBT.1633

Kwon, T., Yoo, W. G., Lee, W. J., Kim, W., & Kim, D. W. (2015). Next-generation sequencing data analysis on cloud computing. In *Genes and Genomics* (Vol. 37, Issue 6, pp. 489–501). Genetics Society of Korea. https://doi.org/10.1007/s13258-015-0280-7

Li, B., & Dewey, C. N. (2011). RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, *12*(1), 1–16. https://doi.org/10.1186/1471-2105-12-323/TABLES/6

Lister, R., O'Malley, R. C., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H., & Ecker, J. R. (2008). Highly integrated single-base resolution maps of the epigenome in Arabidopsis. *Cell*, *133*(3), 523–536. https://doi.org/10.1016/J.CELL.2008.03.029

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.Journal*, *17*(1), 10–12. https://journal.embnet.org/index.php/embnetjournal/article/view/200/479

Mell, P. M., & Grance, T. (2011). *The NIST Definition of Cloud Computing*. https://doi.org/10.6028/NIST.SP.800-145

Mohsen, A., Chen, Y.-A., Allendes Osorio, R. S., Higuchi, C., & Mizuguchi, K. (2022). Snaq: A Dynamic Snakemake Pipeline for Microbiome Data Analysis With QIIME2. *Frontiers in Bioinformatics*, *2*, 63. https://doi.org/10.3389/FBINF.2022.893933

Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021). Sustainable data analysis with Snakemake. *F1000Research*, *10*, 33. https://doi.org/10.12688/F1000RESEARCH.29032.1

Mrozek, D. (2020). A review of Cloud computing technologies for comprehensive microRNA analyses. In *Computational Biology and Chemistry* (Vol. 88). Elsevier Ltd. https://doi.org/10.1016/j.compbiolchem.2020.107365

Palankar, M., Iamnitchi, A., Ripeanu, M., & Garfinkel $ #, S. (n.d.). *Amazon S3 for Science Grids: a Viable Solution?*

Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., Mungall, K., Lee, S., Okada, H. M., Qian, J. Q., Griffith, M., Raymond, A., Thiessen, N., Cezard, T., Butterfield, Y. S., Newsome, R., Chan, S. K., She, R., Varhol, R., … Birol, I. (2010). De novo assembly and analysis of RNA-seq data. *Nature Methods*, *7*(11), 909–912. https://doi.org/10.1038/NMETH.1517

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139. https://doi.org/10.1093/BIOINFORMATICS/BTP616

Stark, R., Grzelak, M., & Hadfield, J. (2019). RNA sequencing: the teenage years. *Nature Reviews Genetics*, *20*(11), 631–656. https://doi.org/10.1038/S41576-019-0150-2

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., & Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology 2010 28:5*, *28*(5), 511–515. https://doi.org/10.1038/nbt.1621

Zhang, X., & Jonassen, I. (2020). RASflow: An RNA-Seq analysis workflow with Snakemake. *BMC Bioinformatics*, *21*(1), 1–9. https://doi.org/10.1186/S12859-020-3433-X/TABLES/2

Zhao, S., Watrous, K., Zhang, C., & Zhang, B. (2017). Cloud Computing for Next-Generation Sequencing Data Analysis. In *Cloud Computing - Architecture and Applications*. InTech. https://doi.org/10.5772/66732

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dr. Gopal Joshi
Circulant Solutions Inc.
gopalj@circulants.com

Satyoki Chatterjee
Circulant Solutions Inc.
satyokic@circulants.com

Pankaj Choudhary
Circulant Solutions Inc.
pankajc@circulants.com

Sanjay Koshatwar
Circulant Solutions Inc.
sanjay@circulants.com

Shekhar Seera
Circulant Solutions Inc.
shekhars@circulants.com