DISEÑO E IMPLEMENTACIÓN DE UN FLUJO DE TRABAJO BIOINFORMÁTICO EN LA NUBE PARA LA IDENTIFICACIÓN DE VARIANTES ONCOGÉNICAS A PARTIR DE DATOS GENÓMICOS

DANIELA VARELA TABARES

Trabajo de grado para optar al título de Ingeniera Biomédica

Nathalia María Vanessa Flórez Zapata, PhD

Juan Esteban Arango Ossa, Ingeniero Biomédico y Mecatrónico

UNIVERSIDAD EIA
INGENIERÍA BIOMÉDICA
ENVIGADO
2019

CONTENIDO

pág.

LISTA DE TABLAS

LISTA DE FIGURAS

GLOSARIO

AWS (AMAZON WEB SERVICES): plataforma que ofrece soluciones flexibles, escalables y confiables de computación en la nube (Amazon Web Services, 2019m).

CLI (COMMAND LINE INTERFACE): interfaz de línea de comandos. Programa que permite la comunicación con el sistema operativo mediante una interfaz de texto (Cortés Martín, 2014).

S3 (SIMPLE STORAGE SERVICE): servicio de almacenamiento escalable, basado en web, diseñado para el archivado de datos y aplicaciones en AWS (Amazon Web Services, 2019l).

EFS (ELASTIC FILE SYSTEM): sistema de almacenamiento de archivos en la nube de AWS (Amazon Web Services, 2019g).

EBS (ELASTIC BLOCK STORAGE): servicio de almacenamiento que hace las veces de disco duro persistente para las instancias EC2 (Amazon Web Services, 2019f).

CIRCOS PLOT: Gráfico de visualización circular para presentar información genómica como alineamientos de secuencia, reordenamientos, expresión génica, niveles de metilación, entre otros (Perrin, 2017).

ECS (ELASTIC CONTAINER SERVICE): servicio de gestión de contenedores escalable y de alta disponibilidad, de AWS, para correr aplicaciones en la nube (Amazon Web Services, 2019b).

EC2 (AMAZON ELASTIC COMPUTE CLOUD): servicio de AWS que ofrece capacidad de computo en la nube (Amazon Web Services, 2019a).

IAM (IDENTITY AND ACCESS MANAGEMENT): servicio que se encarga del manejo de identidades y permisos en AWS (Amazon Web Services, 2019c).

CONTENEDOR: unidad de software que empaca y aísla un conjunto de códigos y sus dependencias para que funcione en diferentes sistemas independiente de la infraestructura (Docker Inc, 2019).

INSTANCIA: es una máquina virtual en un ambiente de computación (FUGA BV, 2019).

PARES DE BASES (pb): en biología molecular, son dos moléculas complementarias de nitrógeno unidas por enlaces de hidrógeno. Se usan como unidad medida, representando el número de nucleótido en una cadena de ADN, siendo cada uno un par de bases (Encyclopædia Britannica, 2019).

SECUENCIA DE EXTREMO EMPAREJADO (PAIRED-END SEQUENCING): método en el que se secuencian ambos extremos de un fragmento de ADN, facilitando el posterior proceso de alineamiento (Illumina, 2019b).

RESUMEN

La secuenciación de alto rendimiento (NGS, por sus siglas en inglés) revolucionó el campo de la genómica al reducir los costos y aumentar la velocidad del proceso drásticamente. Como consecuencia, la cantidad de secuencias de ADN ha aumentado exponencialmente, y cada día se desarrollan nuevas herramientas y aplicaciones para su procesamiento. Esto ha incrementado la demanda de infraestructura de cómputo y almacenamiento, que permita analizar tal volumen de información. El verdadero valor de estos datos se materializa cuando arrojan información médica relevante en escalas de tiempo aceptables, para su aplicación en el diagnóstico y tratamiento de enfermedades asociadas a alteraciones del genoma, como el cáncer.

Sin embargo, la posibilidad de acceder a una infraestructura propia de computación de alto rendimiento, para lograr la transformación de los datos, se ve limitada por sus costos. Es aquí donde la computación en la nube se presenta como una opción atractiva, particularmente por su modelo de facturación bajo demanda, en el que se paga únicamente por los recursos usados.

En este trabajo se expone el proceso para lograr la implementación de un flujo de trabajo bioinformático en la nube para la detección de variantes oncogénicas, desde la selección del proveedor del servicio de computación hasta la validación de las variantes genéticas encontradas. Incluyendo múltiples etapas de levantamiento de requerimientos, codificación, diseño y documentación, como se realiza en las metodologías ágiles para el diseño de software, cuyos principios fueron adoptados para el desarrollo.

Para la infraestructura de computación en la nube, se escogió Amazon Web Services, un proveedor del servicio. Luego se diseñó el flujo, definiendo entradas, procesos intermedios, salidas y herramientas a utilizar en cada paso; y se seleccionaron los archivos de entrada de una base de datos pública. Los diferentes pasos se conectan a través de Isabl, un marco de trabajo para el manejo de datos NGS, que administra los metadatos y gestiona las tareas, desplegado en la nube utilizando el servicio ECS para la orquestación de contenedores. Adicional a este, se utilizó EC2 para el procesamiento y EFS para el almacenamiento, entre otros servicios.

La implementación fue realizada con éxito, se validaron las variantes genéticas encontradas respecto a estudios relacionados y se reportan gráficamente los resultados, lo que facilita la interpretación de estos y le genera un valor adicional al proyecto. Se analizaron los costos asociados al proceso y se comparó con el servicio ofrecido por un centro de computación de alto rendimiento colombiano, mostrando la viabilidad de la computación en la nube para este tipo de desarrollos a corto plazo y pequeña escala.

Palabras clave: flujo de trabajo, computación en la nube, variantes oncogénicas.

ABSTRACT

Next generation sequencing revolutionized the field of genomics by dramatically reducing costs and increasing the speed of the process. Therefore, the number of DNA sequences is exponentially growing, new tools and applications for processing are being developed every day. This has increased the demand for computing and storage infrastructure, which allows analyzing such volumes of information. The significance of these data is materialized when they provide relevant medical information in acceptable time frames, for application in the diagnosis and treatment of diseases associated with genome alterations, such as cancer.

However, the opportunity for researchers to access their own high-performance computing infrastructure is limited by its costs. Hence cloud computing appears as an attractive option, particularly for its on-demand or pay-per-use billing model.

In this work, the process to implement a cloud-based bioinformatic workflow for the detection of oncogenic variants is exposed, from the selection of the computing service provider to the validation of the genetic variants found. Including multiple stages of requirements gathering, coding, design and documenting, as in agile methodologies for software design, which principles were adopted.

AWS was chosen as the service provider for the cloud computing infrastructure. The flow was then designed, defining inputs, intermediate processes, outputs and tools to be used in each step. Input files from a public database were selected. The steps were connected through Isabl, a data science framework for NGS, which manages metadata and tasks. It was deployed in the cloud using ECS for container orchestration. Besides, EC2 was used for processing and EFS for storage, among other services.

The implementation was carried out successfully. The genetic variants found were validated, using related studies, and the results were reported graphically, simplifying the interpretation and generating added value to the project. The costs associated with the process were analyzed and compared with the service offered by a Colombian high-performance computing center, showing the viability of cloud computing for this type of short-term and small-scale developments.

Keywords: pipeline, cloud computing, oncogenic variant.

INTRODUCCIÓN

Las nuevas tecnologías de secuenciación (NGS) han revolucionado las ciencias biológicas. Gracias a ellas ha aumentado la capacidad para secuenciar genomas a mayor velocidad y menores costos. Cada vez hay más datos y herramientas disponibles, la explosión de datos está creciendo desmesuradamente e invade la medicina. El poder acceder a la información específica de los genes junto con la información clínica del paciente, está encaminando la práctica médica hacia una nueva y prometedora forma de ejercerla: la Medicina Personalizada.

Una de las enfermedades más estudiadas desde este enfoque es el cáncer, una enfermedad asociada a los cambios genéticos, bien conocida por su letalidad y diversidad. Entender estos cambios genéticos y perfiles de expresión génica de las células cancerosas permite crear estrategias de tratamiento más efectivas para cada paciente y nuevas formas de diagnóstico y prevención

Aunque el acceso a los datos genómicos representa una oportunidad para la investigación, el recurso computacional necesario para utilizarlos puede ser un obstáculo en muchos casos, porque su potencial no se materializa hasta que sean procesados y arrojen información médica relevante en escalas de tiempo aceptables, lo que requiere gran capacidad de cómputo y almacenamiento.

Frente a esta avalancha de datos y a la dificultad que conlleva hacerse a una infraestructura informática, la computación en la nube ofrece nuevas posibilidades con un modelo bajo demanda, conveniente para los usuarios. Éste es hoy un recurso fundamental para todas las industrias, y su escalabilidad y accesibilidad lo hacen cada vez más atractivo. El reto de los investigadores estará en tener la capacidad para desarrollar flujos de trabajo bioinformáticos escalables e implementables en este tipo de servidores virtuales.

Resulta pertinente y relevante hacer un acercamiento a la implementación de flujos de trabajo en este tipo de ambientes, y que permitan extraer información en gráficas y reportes mucho más comprensibles por el médico, a partir de datos de secuenciación en bruto. Por lo anterior, el objetivo del presente trabajo es implementar un flujo de trabajo bioinformático en la nube para la identificación de variantes oncogénicas a partir de datos de secuenciación masiva.

Para lograrlo, primero se seleccionó AWS como el proveedor del servicio de computación en la nube por medio de una comparación de las diferentes opciones del mercado y el cumplimiento de criterios predefinidos, correspondiente a la primera parte de la metodología. Se siguió con el diseño del flujo, definiendo las entradas, salidas, conjunto de datos y herramientas a emplear en cada paso, teniendo en cuenta que para este trabajo se utilizó Isabl como marco de trabajo para el manejo de datos NGS.

Para la implementación, el desarrollo de códigos y despliegue de toda la infraestructura, se adoptaron las ideas de las metodologías ágiles para el desarrollo de software, que

favorecieron la reestructuración repetitiva del proyecto con ciclos iterativos de trabajo compuestos por análisis de requisitos, diseño, desarrollo, pruebas y documentación. Finalmente, se ejecutaron las operaciones de procesamiento en la nube para cada paso del flujo, a través de Isabl desplegado en la nube, y se evaluaron los resultados.

En la presentación de los resultados, se muestra el proceso de selección del proveedor, el diseño esquemático del flujo, los servicios utilizados durante el despliegue de la infraestructura en la nube, las gráficas de su uso de memoria y almacenamiento arrojadas por AWS y se evalúan los resultados obtenidos.

Se validaron las variantes estructurales detectadas al hacer una revisión de artículos relacionados con la enfermedad específica de la muestra estudiada. Se analizaron el uso de memoria y almacenamiento del proceso, se calcularon los costos de la implementación de acuerdo con los servicios de AWS utilizados y se comparó con los precios de una infraestructura similar en un centro de computación de alto rendimiento.

La nube demuestra ser una opción viable y atractiva para este tipo de desarrollos por sus características de versatilidad, escalabilidad y las soluciones que ofrece según las necesidades del usuario. Se espera que este trabajo sirva como punto de partida para futuros estudios investigativos.

1. PRELIMINARES

1.1 PLANTEAMIENTO DEL PROBLEMA

La era post-genómica, cuyo inicio se relaciona principalmente con la finalización del Proyecto Genoma Humano (HGP, por sus siglas en inglés) hacia el año 2003, ha demostrado que la ciencia, la tecnología y la medicina van de la mano hacia una nueva generación en el área de la salud y del entendimiento de la vida en general (Grubka & Jacobs, 2014). Desde el año 2007, cuando se secuenció por primera vez el genoma completo de un individuo por medio de secuenciación masiva paralela (Wheeler et al., 2008) y se introdujo la secuenciación de alto rendimiento (NGS, por sus siglas en inglés), el panorama de la genómica ha cambiado completamente.

Las nuevas tecnologías han provocado una reducción en los precios de secuenciación que hace unas décadas era impensable. En la Figura 1 se muestra el decrecimiento del costo por genoma en comparación con la ley de Moore, la cual describe una tendencia a largo plazo en la industria de hardware computacional de aumentar la capacidad de cómputo en aproximadamente el doble cada dos años (National Human Genome Research Institute (NHGRI), 2018). Es evidente que el declive en el costo de secuenciar el genoma ya no puede ser descrito por esta ley, y que su caída más abrupta coincide con el momento en que los centros de secuenciación pasaron de los métodos tradicionales de Sanger a la antes mencionada: NGS. Estas plataformas ahora tienen la capacidad de secuenciar aproximadamente 5000 megabases al día a un costo de centavos por megabase (Souilmi et al., 2015).

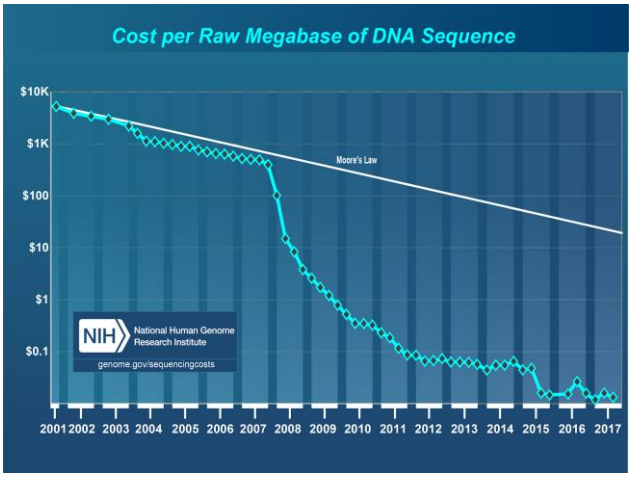

Figura 1. Decrecimiento del costo de secuenciación por genoma a través del tiempo (National Human Genome Research Institute (NHGRI), 2018).

La reducción de precios y el aumento en la velocidad de secuenciación vienen acompañados de una acumulación de grandes cantidades de información en diferentes presentaciones y formatos. Como ejemplo, la secuenciación del genoma humano en una plataforma Illumina HiSeq 2000 genera miles de millones de lecturas cortas con 100 pb (pares de bases) cada una y los archivos donde se almacenan pueden llegar a ser de hasta 460GB (Zhao et al., 2013).

Esta gran explosión de información requiere que ahora la genómica sea tratada desde el enfoque del Big Data, teniendo además en cuenta que esta revolución apenas comienza. Para el año 2025, se espera que la cantidad de datos en el ámbito de la genómica esté a la par o incluso sobrepase los mayores generadores de Big Data actuales: la astronomía, Twitter y YouTube (Stephens et al., 2015); con el agravante de que en ciencias como la genómica el recurso computacional necesario para almacenar la información en bruto es apenas el requerimiento inicial del proceso subyacente. En otras palabras, el potencial de los datos genómicos no se materializa completamente hasta que los datos son procesados y arrojan información médica relevante en escalas de tiempo aceptables, todo esto apuntando hacia la nueva y prometedora forma de ejercer la práctica médica: la Medicina Personalizada o de precisión (Souilmi et al., 2015); en la que se usa información específica de los genes, las proteínas y el ambiente de la persona para el diagnóstico y tratamiento.

Aumentar la precisión y asertividad de las decisiones médicas para mejorar la atención al paciente ha sido siempre uno de los objetivos de la medicina, y está misión se está llevando a un siguiente nivel gracias a nuevas fuentes de información como las pruebas genéticas, las nuevas tecnologías de computación, datos poblacionales, datos del ambiente, y otros, que hacen posible entender la enfermedad desde un punto de vista más personalizado (Xtelligent Healthcare Media, 2018).

Una de las enfermedades más estudiadas y pioneras de este enfoque de la medicina de precisión es el cáncer, una enfermedad bien conocida por su letalidad e incidencia, y que despierta un gran interés por su complejidad (Nagahashi et al., 2019). Entender los cambios genéticos y perfiles de expresión génica de las células cancerosas permite crear estrategias de tratamiento más efectivas para cada paciente y nuevas formas de diagnóstico y prevención (Verma, 2012), beneficios que sólo son una realidad cuando se tienen métodos de procesamiento lo suficientemente robustos para analizar una gran cantidad de datos, y sistemas automatizados para la identificación e interpretación del efecto funcional de las variaciones genéticas en relación con fenotipos específicos (Fernald, Capriotti, Daneshjou, Karczewski, & Altman, 2011; Instituto Nacional del Cáncer, 2019).

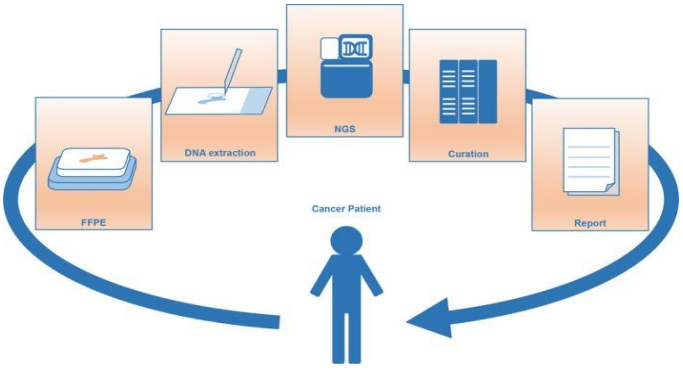

Figura 2. Flujo desde secuenciación de la muestra hasta la interpretación y generación de resultados (Nagahashi et al., 2019).

Para romper la brecha entre la secuenciación y el científico o el médico, no sólo se han publicado los datos, también se han desarrollado un sinnúmero de flujos de trabajo, aplicaciones y herramientas para que la comunidad científica y el público en general

interactúen con ellos (Reid et al., 2014). No obstante, la capacidad de cómputo y almacenamiento que se necesita para llevar a cabo este tipo de análisis sigue siendo un obstáculo, especialmente para investigadores con recursos limitados. El costo computacional y la necesidad de una infraestructura robusta amenazan el desarrollo científico y, en muchos casos, están por encima de lo que muchos laboratorios y centros académicos pueden adquirir (Angiuoli, White, Matalka, White, & Fricke, 2011).

Frente a este panorama y a la dificultades que trae hacerse a una infraestructura informática, la computación en la nube ofrece nuevas posibilidades para analizar datos NGS con un modelo de servicio bajo demanda, atractivo para los usuarios (Angiuoli et al., 2011; Fischer et al., 2012). En este paradigma de computación el procesamiento y almacenamiento sólo existen virtualmente en centros remotos, y se pueden asignar y liberar de forma dinámica según las necesidades particulares (Afgan et al., 2013). Esto presenta a su vez un reto ingenieril y es la capacidad de desarrollar plataformas, aplicaciones y flujos de trabajo escalables e implementables en plataformas en la nube.

Por lo anterior, con el fin de prepararse para un futuro en el que la generación de datos genómicos se convierte en una práctica frecuente en la clínica, es pertinente explorar nuevas posibilidades para su procesamiento y análisis, como la computación en la nube. Para familiarizarse con este recurso y sacarles el máximo provecho a los datos genómicos, se requiere hacer un acercamiento y una exploración sobre la implementación de flujos de trabajo bioinformáticos en la nube y que permiten extraer información importante, como gráficas y reportes, partiendo de datos en bruto adquiridos de bases de datos públicas.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Implementar un flujo de trabajo bioinformático en la nube para la identificación de variantes oncogénicas a partir de datos de secuenciación masiva.

1.2.2 Objetivos Específicos

- Seleccionar un proveedor del servicio en la nube y una plataforma a utilizar por medio de la evaluación y comparación de diferentes opciones en el mercado.

- Diseñar un flujo de trabajo bioinformático para la identificación de variantes oncogénicas, a partir de datos procedentes de secuenciación masiva.

- Implementar las diferentes herramientas que componen el flujo en una instancia en la nube.

- Evaluar el costo-beneficio de la implementación del flujo de trabajo en la nube.

1.3 MARCO DE REFERENCIA

1.3.1 Introducción genómica

ADN (Ácido Desoxi-Ribonucleico): compuesto químico que contiene el material genético de las células. En las células humanas está ubicado principalmente en el núcleo, aunque también hay una pequeña cantidad en la mitocondria (ADN mitocondrial). La información en el ADN se almacena en un código compuesto por las siguientes cuatro bases químicas: adenina (A), guanina (G), citosina (C) y timina (T). El orden de estas es la información necesaria para la creación y mantenimiento de un organismo (NIH- National Human Genome Research Institute, 2015; NIH- U.S National Library of Medicine, 2019b).

Una de sus características más importantes es la capacidad que tiene para replicarse. Cada cadena de la doble hélice que compone su estructura sirve de molde para crear una igual, permitiendo la transmisión de información de generación en generación (Figura 3).

El ADN humano consta de aproximadamente 3 mil millones de bases, y más del 99 por ciento de esas bases son las mismas en todas las personas (NIH- National Human Genome Research Institute, 2015).

Cromosomas: estructuras similares a hilos en las que se empaqueta el ADN en el núcleo de cada célula. Su principal función es compactar la gran cantidad de material genético y organizarlo durante la división celular.

Gen: es un arreglo lineal de nucleótidos localizado en una posición particular del cromosoma que codifica para un producto funcional específico y constituyen la unidad funcional fundamental de la herencia. Cuando un gen está activo, su información se copia primero en otro ácido nucleico, llamado ARN (ácido ribonucleico), que a su vez dirige la síntesis de los productos génicos: las proteínas específicas. Cada persona tiene dos copias de cada gen, una proveniente del padre y otra de la madre. La poca proporción de genes diferentes entre los humanos (menos del 1%) es la responsable de nuestras características únicas (Chao, 2006; NIH- U.S National Library of Medicine, 2019a).

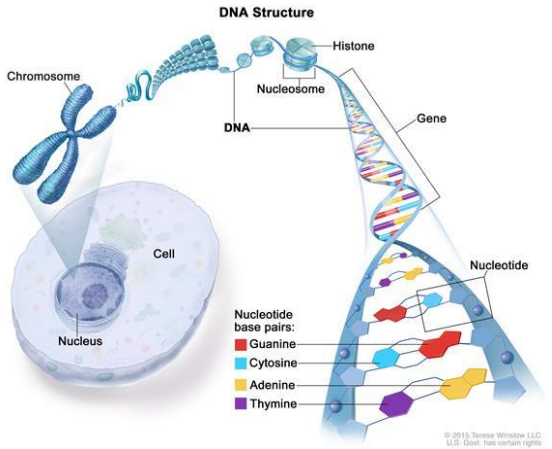

Figura 3. Estructura del ADN (NIH- National Cancer Institute, n.d.).

Genoma: conjunto completo de ADN de un organismo, es decir, el material genético. Incluye tanto las regiones codificantes como las no codificantes (NIH- National Human Genome Research Institute, 2015).

Genómica: estudio de la estructura y función del genoma incluyendo genes y secuencias de ADN que los rodean (Tefferi, 2006).

1.3.2 Variaciones genéticas y el cáncer

Las variaciones son el término general que se usa para indicar diferencias en el ADN en comparación con una secuencia de referencia. Las mutaciones son un tipo de ellas y se definen como alteraciones permanentes en la secuencia de ADN que constituye un gen, es decir, la secuencia difiere de la que se encuentra en la mayoría de las personas (Clancy, 2008; NIH- National Library of Medicine, 2019).

Las variaciones estructurales (SVs, *Structural Variants*) son un tipo de alteraciones del genoma que contribuyen a la diversidad genética y al desarrollo de ciertas enfermedades. Están presentes en regiones del ADN de aproximadamente 1kb o más (National Center for Biotechnology Information (NCBI), 2017).

Hace unos años eran consideradas eventos no muy comunes, pero con el desarrollo de nuevas tecnologías se ha mejorado su detección y son reconocidas ahora como la principal fuente de variación genética que afecta más de un sólo nucleótido.

Este tipo de variaciones pueden ser balanceadas, en las que no se presenta pérdida o ganancia de material genético (inversiones o translocaciones). O desbalanceadas, donde parte del material se pierde o se duplica, y reciben también el nombre de variaciones en el número de copias o CNV (*Copy Number Variations*) (Escaramís, Docampo, & Rabionet, 2015). En la Figura 4 se representan gráficamente estos tipos.

- **Translocación (TRA):** un segmento de ADN cambia de posición dentro del mismo cromosoma o fuera de él.
- **Inversión (INV):** el segmento está en una orientación inversa respecto al resto del cromosoma.
- **Inserción (INS):** aparición de un nuevo segmento respecto a la referencia.
- **CNV:** segmento de ADN presente un número diferente de veces respecto al genoma.
- **Deleción (DEL):** pérdida de material genético respecto a la referencia.
- **Duplicación (DUP):** llamada tándem cuando ocurre inmediatamente después del segmento original.

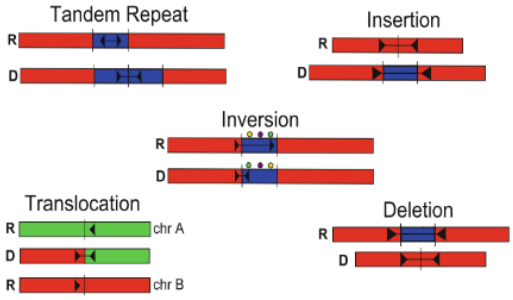

Figura 4. Tipos de variaciones estructurales del genoma (Hayes, 2019).

Cáncer: enfermedad genética, causada por cambios en los genes que controlan la forma en que funcionan las células, especialmente cómo crecen y se dividen. Resulta de la expansión clonal de una sola célula anormal. Las tasas de diferentes procesos mutacionales varían entre los tumores y los tipos de cáncer, pero la mayoría de los cánceres tienen de 1000 a 20,000 mutaciones somáticas y de algunos a cientos de inserciones, eliminaciones y reordenamientos (Martincorena & Campbell, 2015; NIH- National Cancer Institute, 2017).

La aparición y progresión de esta enfermedad es en muchos casos desencadenada por la acumulación de anormalidades en la estructura genómica, como las SVs, a través de diferentes mecanismos como la desactivación de genes supresores de tumores. La genómica del cáncer ha contribuido a la medicina oncológica proporcionando un panorama y un catálogo de mutaciones somáticas presentes en el cáncer humano; información que puede ser utilizada en nuevos enfoques terapéuticos y de diagnóstico (Hayes, 2019; Yi & Ju, 2018).

Muchos grupos de investigación, en especial dos grandes consorcios internacionales (*The International Cancer Genome Consortium* (ICGC) y *The Cancer Genome Atlas* (TCGA)), han analizado conjuntos de datos a gran escala para caracterizar una gran variedad de tipos de tumores comunes y raros. Estas investigaciones han motivado el desarrollo de múltiples algoritmos y herramientas computacionales para una precisa detección y caracterización de SVs (Yi & Ju, 2018).

1.3.3 Bioinformática

La bioinformática es una ciencia que integra datos biológicos con técnicas de almacenamiento, distribución y análisis de la información para apoyar la investigación científica (M. Lesk, 2019). Es relativamente nueva y experimentó un crecimiento explosivo a partir del PGH y los rápidos avances en las tecnologías de secuenciación de ADN (NIH-National Human Genome Research Institute, 2013).

Secuenciar es determinar el orden exacto de las bases en una cadena de ADN. Como las bases existen por pares, basta con obtener la secuencia de una de las cadenas (NIH-National Human Genome Research Institute, 2015).

La secuenciación de nueva generación (NGS), también conocida como secuenciación de alto rendimiento, se refiere al conjunto de tecnologías modernas que permiten secuenciar

el ADN y RNA de una forma mucho más rápida y económica que la originalmente usada secuenciación Sanger (basada en el proceso biológico de la replicación del ADN) o similares. Estos nuevos métodos utilizan procesos paralelos masivos, y como ejemplo, se encuentran: secuenciación Illumina (Solexa), secuenciación Roche 454, secuenciación Ion torrent: Proton/PGM y secuenciación SOLiD (Illumina, 2019a).

La gran ventaja que presentan es la escalabilidad, permitiendo secuenciar todo el genoma a la vez por medio de su división en pequeños fragmentos que se secuencian por separado de forma automática, para luego ser alineados en una gran secuencia (EMBL-EBI, n.d.; Hewlett Packard Enterprise Development LP, 2019; Straiton, Free, Sawyer, & Martin, 2019).

Tipo de secuenciación del ADN (Illumina, 2019a):

- La secuenciación completa del genoma (WGS, *Whole-Genome Sequencing*): método integral para el análisis de genomas completos.
- Secuenciación dirigida (*targeted* - "*hot-spot" sequencing panel*): secuenciación de una región aislada del genoma.

NGS genera grandes cantidades de datos que requieren múltiples pasos computacionalmente intensivos para que se realice un análisis apropiado. Al conjunto de algoritmos bioinformáticos que se ejecutan en una secuencia predeterminada para procesar los datos se le llama Flujo de trabajo o "pipeline" (Roy et al., 2018).

La mayoría de los análisis de NGS comparten una serie de pasos en común, Figura 5, por lo que se han establecido unos flujos de trabajo altamente estandarizados, pero a la vez lo suficientemente flexibles para acoplarse al objetivo específico del proyecto (Kulkarni & Frommolt, 2017).

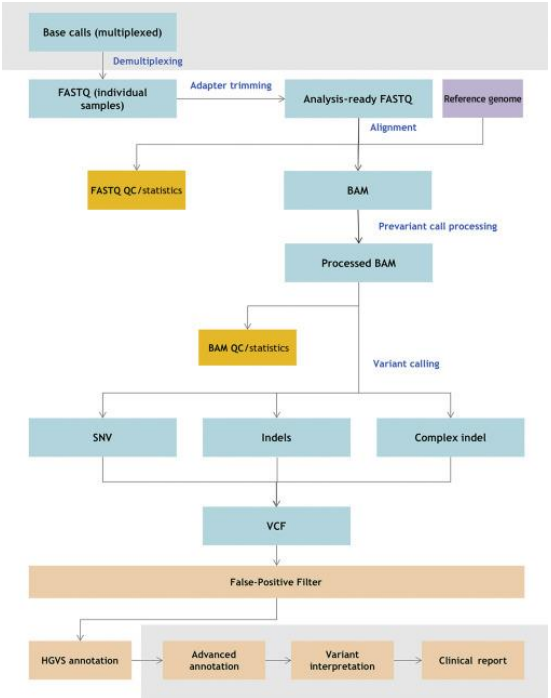

Figura 5. Componentes típicos de un flujo de trabajo bioinformático usando NGS (Roy et al., 2018).

A continuación se describen los pasos fundamentales y sus archivos de salida, que concuerdan con los de la figura anterior.

a. **Generación de secuencias**: se convierten las señales del sensor de la plataforma en secuencias de nucleótidos con su respectivo puntaje de calidad. Se almacena en archivo FASTQ.

b. **Alineamiento**: se determina dónde cada secuencia corta se alinea con el genoma de referencia. Las salidas del secuenciador son lecturas cortas (<250 pb) aleatoriamente distribuidas. Es un proceso exigente en términos computacionales, que asigna una puntuación de calidad de mapeo para cada lectura corta, indicando la confidencia del proceso. El resultado se entrega generalmente en formato BAM (*Binary Alignment Map*) que es una versión binaria del formato SAM (*Sequence Alignment and Mapping*) (Roy et al., 2018).

   Un ejemplo de la cantidad de secuencias que se pueden llegar a alinear es la salida de una máquina como la The Illumina/Solexa, que únicamente en una corrida arroja entre 50 y 200 millones de secuencias cortas (32–100 bp). Para llevar a cabo este proceso de una forma eficiente y acertada, se han desarrollado muchos métodos y algoritmos de alineación con diferentes principios. Algunos de los programas de alineación más utilizados son. MAQ, SOAP, Bowtie, BWA, BLAST, SeqMap, entre otros (H. Li & Durbin, 2009).

c. **Llamado de variantes**: es el proceso para la identificación de las diferencias o variaciones entre la muestra y la secuencia del genoma de referencia. La entrada es comúnmente un archivo en formato BAM o similares, y la salida se presenta en un archivo de tipo VCF (*Variant Call Format*), cuyas características dependen del tipo de variante (Roy et al., 2018).

   Dentro de los métodos de detección de variantes basados en secuenciación masiva existen una gran cantidad de algoritmos, que varían en cuanto a la sensibilidad y especificidad dado que usan diferentes características de las lecturas cortas para llevar a cabo el proceso, por lo que pueden arrojar resultados inconsistentes para la misma muestra. El ruido de los datos o la presencia de regiones poco caracterizadas del genoma también pueden afectar la detección. Por esto, aunque en teoría pueden detectar cualquier tipo de SVs, ninguno de ellos es capaz de identificar de manera exacta SVs de todos los tipos y todos los tamaños. Es recomendable para cualquier proyecto utilizar más de un algoritmo y unir las salidas para incrementar la precisión en los resultados (Guan & Sungab, 2016; Kosugi et al., 2019).

d. **Filtrado y anotación**: este paso puede ser opcional o variar según los objetivos de la investigación. Se identifican falsos positivos o se eliminan variantes que no sean de interés según filtros establecidos. Posteriormente se realizan comparaciones de las variantes encontradas respecto a bases de datos para su caracterización y que puedan ser interpretas clínicamente (Roy et al., 2018).

Cuando se hace un análisis bioinformático para datos genómicos, es importante tener en cuenta que existe una considerable cantidad de marcos, aplicaciones, herramientas,

sistemas y demás tecnologías, que tienen clasificaciones diferentes dependiendo de su función y la escala en la que operan. Es fundamental diferenciar a qué parte del proceso y a qué necesidad responden para poder compararlos correctamente, por lo que se hace una breve clasificación para enmarcar el que usa en este proyecto. En la Tabla 1 se presentan algunos ejemplos conocidos de acuerdo con cada clasificación.

- **Sistema de Gestión de Análisis de Información (*Analysis Information Management Systems*, AIMS)**: marcos que permiten integrar metadatos, estructurar la información, procesar pipelines automáticamente, manejar las operaciones desde un sistema centralizado, entre otros. Cuentan con la creación de una base de datos y la capacidad de implementar operaciones sistemáticamente utilizando diferentes paradigmas de cómputo y almacenamiento (Griffith et al., 2015a).

- **Sistema de gestión del flujo de trabajo (*Workflow Management Systems*, WMS)**: software que proporcionan la infraestructura o ambiente, y las herramientas para diseñar, configurar, ejecutar y monitorear flujos de trabajo (The Apache Software Foundation, 2018).

- **Plataforma bioinformática**: es un concepto de software más general, utilizado en el campo de la bioinformática para referirse a la prestación de PaaS para el manejo y análisis de los datos. Es un servicio más completo, robusto y automatizado. Generalmente es una infraestructura que funciona en la nube (Seven Bridges Genomics, 2019).

Tabla 1. Ejemplos de herramientas según su clasificación.

Concepto	Ejemplos
AIMS	GMS (*Genome Modelling System*) (Griffith et al., 2015b) SeqWare (O'Connor, 2014). QuickNGS (Wagle, Nikolić, & Frommolt, 2015). HTS-flow (Bianchi et al., 2016). Isabl (Medina-Martínez et al., 2019).
WMS	Bpipe (Sadedin, Pope, & Oshlack, 2012). Taverna (Wolstencroft et al., 2013). COSMOS (Gafni et al., 2014).
Plataforma	Galaxy (Afgan et al., 2018). DNAnexus (DNANEXUS, 2019). Seven Bridges (Seven Bridges Genomics, 2019). FireCloud (Broad Institute, 2019).

Fuente: Autor del trabajo.

En el laboratorio Papaemmanuil del Memorial Sloan Kettering Cancer Center (MSKCC) se han explorado diferentes herramientas bioinformáticas, tanto de código abierto como desarrolladas por el mismo equipo de trabajo. Como parte de la experiencia adquirida como practicante de esta institución, se trabajó con Isabl, un marco de trabajo para datos NGS creado por los ingenieros del laboratorio, que a su vez es un AIMS. En este se observó un

amplio potencial para su implementación en plataformas IaaS o en cualquier ambiente, y por lo tanto fue seleccionada para el desarrollo de este trabajo. Esta investigación podrá servir también como caso de estudio para este desarrollo emergente, que se encuentra en proceso de publicación (Medina-Martínez et al., 2019).

Isabl cuenta con una infraestructura modular compuesta por cuatro componentes, Figura 6:

- Isabl-db: base de datos relacional con un modelo centrado en el individuo.
- Isabl-api: RESTful API (*Application Programming Interface*) para la integración del procesamiento de datos con sistemas empresariales.
- Isabl-cli: línea de comandos para manejar los datos e implementar aplicaciones bioinformáticas.
- Isabl-web: aplicación web.

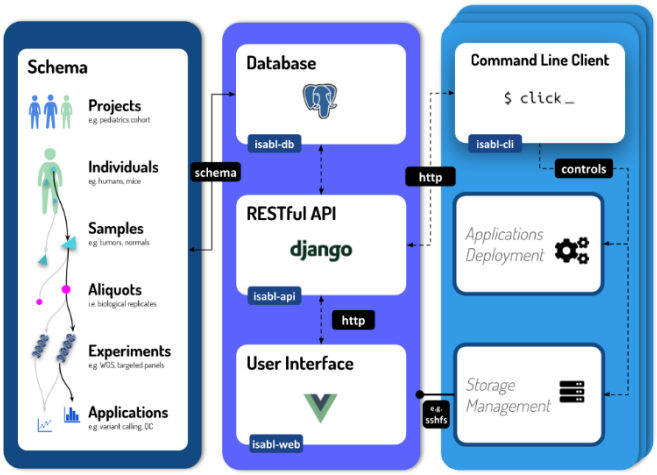

Figura 6. Representación de la arquitectura y el modelo de metadatos de Isabl (Medina-Martínez et al., 2019).

Isabl permite el monitoreo y procesamiento de los datos a escala, por medio de su importación recursiva y la ejecución de aplicaciones o flujos de trabajo propios del usuario basado en las conexiones de los metadatos, con comandos simples. No es un PaaS ni un sistema de manejo de flujo, en ella se pueden registrar cualquier tipo de aplicaciones y herramientas, sin importar la tecnología para su desarrollo y el ambiente en el que se corran (Medina-Martínez et al., 2019).

1.3.4 Computación en la nube

Computación de alto rendimiento- HPC (High Performance Computing): se refiere a la práctica de agregar potencia de cómputo con el fin de obtener la habilidad de procesar datos, almacenarlos y realizar complejos cálculos a grandes velocidades (NetApp, 2019). Un sistema HPC es una red de nodos, cada uno con uno o más chips de procesamiento, así como su propia memoria (National Institute for Computational Sciences, n.d.).

Computación en la nube (Cloud Computing): la computación en nube hace referencia al ofrecimiento de servicios de computación desde servidores en una red (SSH Communications Security, 2018). Es un modelo de servicio que permite el acceso

conveniente y bajo demanda a un conjunto de recursos informáticos compartidos y configurables como redes, servidores, almacenamiento, aplicaciones, entre otros; que pueden usarse y liberarse con un mínimo esfuerzo de administración o interacción entre las partes (Mell & Grance, 2011).

Los tipos o modelos en los que se puede desplegar la infraestructura en la nube son los siguientes (Mell & Grance, 2011; Nexica - Econocom Group, 2015; Visma AS, 2019):

- **Privado**: La infraestructura se proporciona para uso exclusivo de una sola organización que comprende varios consumidores.
- **Público**: La infraestructura está prevista para uso abierto por el público en general.
- **Híbrido**: dos o más infraestructuras en la nube distintas que siguen siendo entidades únicas, pero están unidas por una tecnología que permite la portabilidad de datos y aplicaciones. Para usuarios que tienen infraestructura propia pero buscan aprovechar las ventajas de un proveedor externo.
- **Comunidad**: uso exclusivo por parte de una comunidad específica de consumidores de organizaciones que tienen requerimientos compartidos.

Dentro de tipos anteriores, existen tres modelos principales de informática en la nube, es decir, los servicios ofrecidos pueden dividirse así:

- **Software como servicio (SaaS)**: el consumidor utiliza las aplicaciones del proveedor que residen en la nube remota y se accede a través de la Web o de una API, bajo un modelo de suscripción generalmente. Los usuarios no tienen que administrar, instalar o actualizar software (Barabas, n.d.). En el caso de la bioinformática, SaaS facilita el acceso remoto a las herramientas desarrolladas a escala de nube disponibles a través de Internet, eliminando la necesidad de instalación local y mantenimiento (Z. Zhang, Lin, Xin, Yan, & Jingfa Xiao, 2012)

  Ejemplos: Gnomics, BGI Cloud, EasyGenomics

- **Plataforma como servicio (PaaS)**: proporciona a los usuarios un entorno de nube en el que pueden desarrollar, gestionar y montar aplicaciones. Pueden usar herramientas creadas previamente para personalizar y probar sus propias aplicaciones. El consumidor no administra ni controla la infraestructura de la nube, tampoco la red, los servidores, los sistemas operativos o el almacenamiento (Barabas, n.d.; Mell & Grance, 2011).

  Es especialmente útil porque los recursos se escalan automática y dinámicamente para adaptarse a la demanda de la aplicación. Normalmente incluye entornos de ejecución de lenguaje de programación, servidores web y bases de datos (Z. Zhang et al., 2012).

  Ejemplo: Google App Engine. Eoulsan y Galaxy Cloud para bioinformática.

- **Infraestructura como servicio (IaaS)**: el usuario tiene control sobre los sistemas operativos, el almacenamiento y las aplicaciones; y un posible control limitado sobre los componentes de red seleccionados. En general, son ambientes virtualizados que

proveen todo lo que se requiera una red local, sin los costos de comprar y mantener su propio hardware (Mell & Grance, 2011; tecnoiver ltda, 2015).

Ejemplos: Cloud BioLinux, una máquina virtual pública para computación bioinformática y CloVR, una máquina virtual portátil que incorpora varios canales para el análisis de secuencias (Z. Zhang et al., 2012).

- **Proveedores del servicio de computación en la nube**: son organizaciones que controlan grandes cantidades de computadores y dispositivos de almacenamiento de información, organizados en centros de datos alrededor del mundo. Es al proveedor es a quien el cliente le solicita los recursos y los retorna nuevamente cuando se completa el trabajo. Actualmente existe una gran variedad de ellos, con distintas especialidades y variaciones en los servicios; es un mercado creciente y en proceso de diversificación (Langmead & Nellore, 2018).

2. METODOLOGÍA

2.1 SELECCIÓN DEL PROVEEDOR

Para seleccionar el proveedor de servicio en la nube, se realizó una revisión sobre algunas características y productos ofrecidos por Amazon, Google y Microsoft por ser los tres proveedores comerciales más usados globalmente y líderes del mercado como se expone en múltiples estudios y reportes (Bala, Gill, Smith, & Wright, 2019; Hille & Baum, 2018; Hille, Klemm, & Lemmermann, 2017). En la Figura 7 se evidencia lo anterior, al mostrar a estas tres compañías como las líderes en las diferentes categorías analizadas dentro del mercado de computación en la nube. Se debe tener en cuenta que para esta aplicación, se usa el modelo de servicio IaaS y se buscan productos que respondan a las necesidades de capacidad de cómputo, almacenamiento y manejo de contenedores principalmente.

Figura 7. Comparación de proveedores que muestra a AWS, Google y Microsoft como líderes del mercado de computación en la nube (Hille & Baum, 2018).

De forma paralela se hizo un resumen en forma de tabla de las características principales de algunos flujos de trabajo ya implementados en la nube con el fin de recopilar la información presentada en los antecedentes y tener en cuenta herramientas o servicios que han sido utilizados previamente por otros investigadores para la selección.

Con el fin de comparar las ventajas y desventajas de cada uno de los proveedores, se definieron algunos criterios para asegurarse de hacer una buena selección. Los criterios definidos se basan en estudios comparativos reportados previamente (Hille & Baum, 2018; Islam & Rehman, 2013; Q. Zhang, Cheng, & Boutaba, 2010), y los intereses particulares de la presente implementación. Los criterios seleccionados fueron:

- **Costo**: se tienen en cuenta los créditos gratuitos ofrecidos para cualquier usuario y los que se ofrecen para estudiantes particularmente. Además, se presenta la estimación de costos realizada a través de la página web para una implementación de desarrollo.
- **Escalabilidad**: posibilidad de utilizar la misma infraestructura en la que se desarrollará esta implementación, pero con datos a gran escala, únicamente escalando los recursos, y la facilidad para realizarlo.
- **Administración de los recursos**: diversidad de interfaces para el monitoreo, gestión y configuración de los servicios, y la facilidad para usarlas.
- **Facilidad de implementación**: documentación clara para cada parte del proceso, existencia de simuladores en línea, tutoriales y soporte.
- **Experiencias en bioinformática**: uso en flujos de trabajo bioinformático o proyectos similares. Para esto se hizo una recopilación de la información presentada en los antecedentes del anteproyecto para resumir las herramientas y servicios utilizados por cada uno de los artículos publicados.
- **Seguridad**: cumplimiento de estándares como ISO 27000 y certificaciones como X.509 SSL, uso de protocolos de encriptación como SSL/TLS (Secure Socket Layer/Transport Layer Security) y otros métodos de seguridad de acceso como PKI (Public Key Infrastructure) (Höfer & Karagiannis, 2011). Además, confidencialidad para el acceso y la transferencia de datos, y de auditabilidad (Q. Zhang et al., 2010).

Para el caso del tema clínico es fundamental que se proteja la información médica de los pacientes (PIH, Protected Health Information) por medio de leyes como la Ley de Transferencia y Responsabilidad de Seguro Médico (HIPPA, por sus siglas en inglés). Los proveedores de computación en la nube deben proveer este tipo de protección.

Luego de establecer los criterios se define una ponderación para cada uno de acuerdo con el interés y alcance de la presente implementación. Para la calificación se tuvieron en cuenta experiencias previas, estudios comparativos (Hille & Baum, 2018; Hille et al., 2017; Höfer & Karagiannis, 2011; A. Li, Yang, Kandula, & Zhang, 2010; Q. Zhang et al., 2010), información encontrada en las respectivas páginas web y reuniones con personas experimentadas en el tema de la computación de alto rendimiento.

Entre los expertos consultados se encuentra Juan David Pineda Cárdenas, coordinador técnico de Apolo, centro de computación de la Universidad Eafit, con quien se tuvieron dos reuniones para recibir sugerencias y solucionar dudas sobre los factores que se deben tener en cuenta a la hora de seleccionar el proveedor del servicio y correr análisis a gran escala. También, Martín Elías Quintero Osorio, estudiante de Ingeniería de sistema de la Universidad de Antioquia e ingeniero de aprendizaje de máquinas en guane Enterprises (Ruta N) con quien se tuvieron alrededor de cuatro reuniones cortas para llevar a cabo ensayos con los servicios de AWS y compararlos con los que ofrece Google gracias a su experiencia trabajando con este último.

Se utilizó una gráfica de radar (o diagrama de araña), como alternativa para la visualización de información multivariable, porque permite comparar gráficamente dos o más entidades respecto a múltiples características (InfoSoft Global Private Limited, 2019).

2.2 LEVANTAMIENTO INICIAL DE REQUERIMIENTOS

Teniendo en cuenta las características de Isabl, se hizo un levantamiento preliminar de requerimientos para su implementación en la nube, que sirvió para filtrar dentro de la amplia oferta de Amazon e identificar los servicios de AWS que podrían satisfacerlos.

- Docker y Docker-Compose para las aplicaciones y para Isabl.
- Ambiente con Python 3.6 para la línea de comandos y librerías de soporte.
- ECS, EKS (*Elastic Kubernetes Service*) y/o ECR (*Elastic Container Registry*) para la gestión de contenedores.
- S3 o EFS, para el almacenamiento de datos de entrada, referencias y resultados.
- EC2, como servicio web que proporciona capacidad informática escalable en la nube.

2.3 DISEÑO DEL FLUJO DE TRABAJO

2.3.1 Diseño esquemático y selección de aplicaciones

De acuerdo con los pasos generales de un pipeline (Roy et al., 2018), que se expusieron en el marco de referencia, y el alcance del presente proyecto, se hizo un diseño esquemático en el que se muestran los tipos de archivo de entrada y salida para cada paso y el nombre del proceso entre uno y otro.

La selección de las aplicaciones que cumplieran con los pasos especificados anteriormente se basó en experiencias previas en el área y estudios de su desempeño.

- **Alineamiento**: se utiliza BWA (*Burrows–Wheeler Alignement*) por ser uno de los más utilizados por los diferentes pipelines revisados y porque según estudios como el publicado por H. Li & Durbin, 2009, en el que se compara el rendimiento de diferentes programas, se encuentra entre los más recomendados, fáciles de implementar y con menos requerimientos de memoria (Shang et al., 2014).

  Para esta implementación se usa BWA-MEM (*Maximum Exact Matches*), uno de los tres algoritmos que constituyen BWA, por ser el más reciente y recomendado por su velocidad y precisión.

- **Llamado de variantes**: la preselección de los “llamadores de variantes” (*variant callers*) se basó en un estudio publicado este año: “*Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing*” (Kosugi et al., 2019); por ser la evaluación más completa de este tipo realizada hasta la fecha. En este se evaluaron y compararon 69 algoritmos de detección de SVs usando datos reales y simulados, y se encontró que hay algoritmos más adecuados para cada tipo

y tamaño de SV. Dentro de los resultados obtenidos, se muestran las estadísticas para la precisión (valor-F) de los algoritmos analizados para detectar DELs, DUPs, INSs, e INVs, Figura 8. Se destaca, por la altura de sus barras y variedad de los colores, un subconjunto de ellos capaces de llamar variantes de diversos tipos con gran precisión: 1-2-3-SV, DELLY, GRIDSS, inGAP-sv, Lumpy, Manta, MetaSV, Pindel, SoftSV, SvABA, Wham.

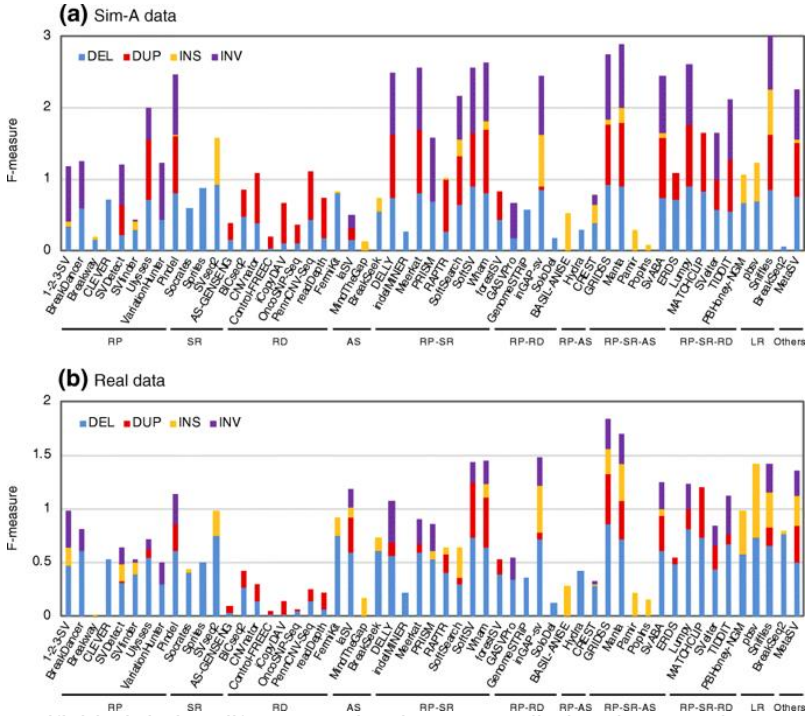

Figura 8. Especificidad de los diferentes algoritmos estudiados de acuerdo con el tipo de SV diferenciado por los 4 colores. Los algoritmos se categorizan en la parte inferior de cada gráfico de acuerdo con el método de detección que utilizan (*RP, read pairs; SR, split reads; RD, read depth; AS, assembly; LR, long reads*) (Kosugi et al., 2019).

Dentro de este subconjunto se escogieron dos algoritmos: GRIDSS (Cameron et al., 2017) y DELLY (Rausch et al., 2012), con los que se había trabajado anteriormente. Ambos estuvieron entre los mejores en cuanto a la precisión para determinar los puntos de ruptura (*breakpoints*) de todos los tamaños. Se tuvo también en cuenta la velocidad de procesamiento y requerimiento computacional, así como la facilidad de implementación de acuerdo con su documentación y la existencia de contenedores Docker en el repositorio Docker Hub para su uso.

- **Filtrado y unión**: el filtrado de variantes se hizo mediante un script de Python. Y la unión de los resultados obtenidos con DELLY y GRIDSS después del filtrado, se hizo mediante SURVIVOR (Jeffares et al., 2017), un conjunto de herramientas para la manipulación de SVs que incluye una función "*merge*". Esta última se descarga fácilmente sin instalaciones adicionales y se encuentra bien documentado (Sedlazeck, 2019).

- **Anotación**: se utiliza svanno, un paquete desarrollado por el laboratorio Papaemmanuil (Elli Papaemmanuil Lab, 2019) al que se le hicieron pequeñas modificaciones. Esta anotación se hace respecto a OncoKB, una base de conocimiento oncológica que contiene información sobre los efectos y las implicaciones en el tratamiento de alteraciones genéticas cancerígenas (Chakravarty et al., 2017).

- **Visualización**: dado que el tamaño de los archivos después de todos los pasos de procesamiento anteriores es lo suficientemente pequeño para ser manipulado localmente (aproximadamente 100kB), la creación de circos plot y el reporte con sus resultados se hace a través de scripts locales en los que se usa Python, HTML y R (paquete RCircos y bedtools).

2.3.2 Búsqueda y selección de datos

Se utiliza el tipo de secuenciación dirigida, principalmente por el gran volumen de información que representa el genoma completo y la restricción del presupuesto. Este tipo de secuenciación presenta ciertas ventajas como la reducción de costos y tiempo en el manejo de los datos, y permite altos niveles de confiabilidad en la secuenciación para la posterior detección de variantes (Illumina, 2019c). Aun así, el flujo implementado puede ser implementado en datos del genoma completo, por medio de la escalación de recursos

Inicialmente se hicieron pruebas con la última versión disponible del genoma en NCBI: GRCh38 (National Center for Biotechnology Information, 2019); sin embargo, los resultados obtenidos eran en su mayoría incompatibles con las aplicaciones y requeriría pasos adicionales para adecuarlos. Por lo anterior, para la versión final del flujo se utiliza la versión GRCh37 (hg19) (University of California, 2018).

Para la descarga de los datos de secuenciación en formato FASTQ se utilizó la base de datos del The European Nucleotide Archive (ENA) y se utilizaron las palabras claves "*Targeted sequencing cancer*" para la búsqueda. Dentro de los resultados encontrados, se escogió el siguiente estudio.

PRJNA299807 (Memorial Sloan Kettering Cancer Center, 2019b): *Targeted sequencing of adenomyoepithelioma*. Los datos fueron secuenciados con la técnica IMPACT, una prueba de secuenciación dirigida disponible para pacientes del MSKCC (Memorial Sloan Kettering Cancer Center, 2019a); con la que se había trabajado antes y se conoce la calidad. Se tomaron las muestras SAMN04215521 y SAMN04215522, identificadas como AM9_Normal_IMPACT y AM9_Tumor_IMPACT respectivamente. Cuenta con una publicación asociada (Geyer et al., 2018).

Los archivos tomados son de tipo FASTQ de extremo emparejado (paired-end) de una muestra normal y otra con tumor, correspondientes a un solo paciente.

2.4 IMPLEMENTACIÓN

Para la implementación del flujo de trabajo usando Isabl, se hicieron ciclos cortos de planeación, análisis de requisitos, diseño, desarrollo, pruebas y documentación, de forma similar a como se realiza en las metodologías ágiles para el desarrollo de software (Guru99, 2019).

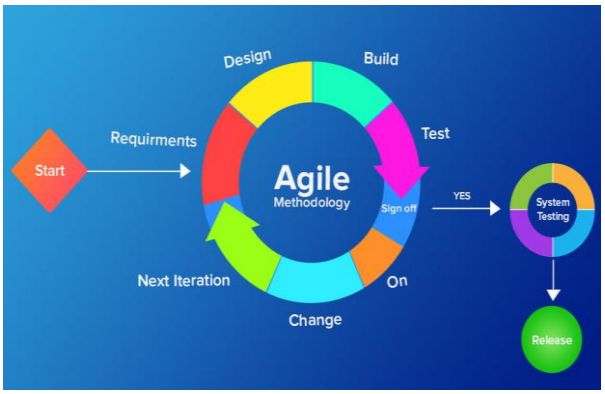

Figura 9. Esquema de la metodología Agile (Singh, 2018)

En este caso se dividió el proyecto en iteraciones de acuerdo con el ambiente de desarrollo y los pasos del flujo, así:

- Tutorial de Isabl con datos de prueba y contenedores.
- Isabl en ambiente local con datos de prueba, codificación del CLI y pruebas de aplicaciones.
- Despliegue de Isabl en la nube.
- Codificación y registro de las aplicaciones, instalación del CLI en la instancia.
- Lanzamiento de la instancia final y ejecución de las aplicaciones.

Inicialmente se hizo un tutorial de Isabl siguiendo la guía "*10 Minutes to Isabl*" (Medina-Martínez et al., 2019) en una máquina local, con Linux, para familiarizarse con la herramienta, siguiendo rigurosamente el paso a paso y pruebas de los comandos. Los requerimientos de este tutorial se encuentran en contenedores Docker.

El paso siguiente fue la instalación de Isabl localmente en un ambiente propio. En esta etapa se definen los requerimientos para la implementación de las aplicaciones de alineamiento y llamado de variantes tras hacer ensayos locales de su desempeño con los mismos datos de prueba utilizados para el tutorial, se hacen modificaciones a su código para adecuarse al ambiente, y se documenta el paso a paso, haciendo especial énfasis en las características y configuraciones propias de Isabl.

Posteriormente, se llevó a cabo el ciclo correspondiente al montaje en un ambiente de nube. Dado que Isabl es un proyecto de Django que utiliza Docker, fue desplegado en AWS usando Amazon ECS para el manejo de contenedores. ECS tiene dos tipos de lanzamiento de tareas: Fargate y EC2 (Amazon Web Services, 2018). El primero elimina la necesidad de administrar y escalar los clúster o servidores, mientras que en el segundo se debe administrar el clúster de instancias para ejecutar los contenedores. Como primera

aproximación se hicieron pruebas utilizando Fargate, pero se decidió cambiar el enfoque hacia EC2 porque se quería tener más manejo sobre los recursos de cómputo, acceso a las instancias y monitoreo de ellas directamente (Amazon Web Services, 2019j).

Una vez se tuvo Isabl funcionando y estable en la nube, se crea otra instancia de forma paralela para el procesamiento. Es decir, Isabl se despliega en su propia instancia de contenedores creada a partir de ECS en el paso anterior y, adicionalmente, se lanza una instancia EC2 para las operaciones, donde se instala el CLI y se administran los demás archivos necesarios. Las instancias de contenedores (CI, *Container Instances*) son instancias Amazon EC2 pero que han sido creadas a través de un clúster ECS y ejecutan el agente de contenedor de Amazon ECS (Amazon Web Services, 2019j).

Inicialmente se hicieron pruebas del funcionamiento con una instancia t2.xlarge para el procesamiento y un *bucket* en S3 con los datos de prueba, pero se observó que la arquitectura y comandos de Isabl se ajustaban más al sistema EFS por lo que se creó un sistema de archivos y se montó en la instancia EC2.

Durante este proceso de pruebas se definieron las instalaciones o contenedores Docker necesarios para cada paso del flujo, se hicieron los ajustes necesarios en el código para que todas las herramientas funcionen correctamente en las instancias y se registraron las aplicaciones a través del API de Isabl. Debido a la posibilidad de escalamiento de la instancia de procesamiento, los repositorios clonados desde Github se almacenaron en el punto de montaje EFS para poder ser compartidos entre múltiples instancias independientemente de su estado.

Después de familiarizarse con el funcionamiento de las instancias y su interacción con Isabl, se hizo un proceso final de levantamiento de requerimientos de memoria para la ejecución de las aplicaciones con los datos reales, por medio de la revisión de la documentación de BWA, GRIDSS y DELLY. Se encontró que los requerimientos mínimos se establecían por el paquete GRIDSS, que sugiere disponer de un mínimo de 16 GiB de memoria (Papenfuss Lab, 2019). Por ello, se escala a la instancia adecuada.

Finalmente, se ejecutan todas las aplicaciones en su orden hasta la anotación y unión de variantes, donde se obtienen archivos de menor tamaño que son descargados y se realiza el reporte localmente sin tener que hacer instalaciones adicionales en la máquina virtual de Amazon. El reporte final fue subido a S3 para facilidad de acceso.

2.5 EVALUACIÓN DE RESULTADOS

2.5.1 Validación de variantes

Para la validación de las variantes oncogénicas encontradas tras el último paso del flujo, se hizo una breve revisión del artículo base del estudio PRJNA299807: *Recurrent hotspot mutations in HRAS Q61 and PI3K-AKT pathway genes as drivers of breast adenomyoepitheliomas* (Geyer et al., 2018) y de otros artículos científicos, donde se reporte la relación de cada gen con el cáncer de mama, por ser esta la enfermedad que se estudia en las muestras seleccionadas.

2.5.2 Análisis del costo de implementación

Se utiliza la herramienta de exploración de costos de AWS, para detallar la facturación por servicios y comprobar su modelo bajo demanda. Allí se incluyen todos los gastos de la implementación, incluyendo pruebas y ajustes.

Para lograr un acercamiento al costo efectivo de implementación, se realiza un cálculo basado en el uso real, excluyendo pruebas y tiempos muertos de las instancias. La aproximación del gasto para los servicios que se cobran como GiB por mes (*per GB-month*), se calculó proporcionalmente según el tiempo de utilización respecto a un mes completo, como lo hace AWS, por medio de su facturación por segundos (Amazon Web Services, 2017).

Debido a las diferencias que existen en la forma en que se facturan los servicios de computación, de acuerdo no sólo al proveedor sino al ambiente desde el que se presten, se hace necesario normalizar los costos para poder realizar una comparación aceptable entre un centro de computación de alto rendimiento y un proveedor de infraestructura en la nube.

Para AWS, se utiliza la herramienta Calculadora de Precios (Amazon Web Services, 2019h), y por otro lado se obtuvo una cotización de los precios por mes actuales de BIOS, Centro de Bioinformática y Biología Computacional de Colombia (BIOS, 2019). BIOS factura por mes de disponibilidad, mientras AWS lo hace por segundos de uso real. Por ello, se ajustan los valores a una facturación mensual equivalente. Se compararon instancias ofrecidas por ambos con características técnicas equivalentes con el fin de realizar una comparación más precisa.

3. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS

3.1 SELECCIÓN DEL PROVEEDOR

Al explorar el repertorio de servicios de Amazon, Google y Microsoft se encuentra una amplia oferta de soluciones, por lo que se resumió la revisión realizada en la Tabla 2, clasificando los productos de acuerdo con su categoría en cada uno de los proveedores.

En cuanto al manejo de contenedores, se presentan tres opciones de servicio, empezando por Kubernetes que es la herramienta de orquestación de contenedores más completa, hasta el servicio de registro que funciona de manera similar en los tres casos.

Para el caso del almacenamiento, dentro de la amplia oferta, se encontraron tres modelos principales: almacenamiento basado en objetos, almacenamiento en bloques y almacenamiento de archivos o ficheros; en este orden se encuentran en la tabla empezando de arriba hacia abajo.

Tabla 2. Resumen de servicios ofrecidos por los proveedores en las áreas de interés.

PROVEEDOR	PLATAFORMA - SERVICIO	Iaas	Contenedores	Almacenamiento	Servidores Virtuales
GOOGLE	Google Cloud Platform	Google Compute Engine	Kubernetes Engine (GKE)	Cloud Storage	Instancias VM
			Instancias Container-optimized	Persistent Disk	
			Container Registry	Cloud Filestore	
AMAZON	amazon web services	Amazon EC2	Amazon Elastic Kubernetes Service (Amazon EKS)	Simple Storage Service (Amazon S3)	VMs
			Amazon Elastic Container Service (ECS)	Elastic Block Store (Amazon EBS)	
			Amazon Elastic Container Registry (ECR)	Elastic File System (Amazon EFS)	
MICROSOFT	Microsoft Azure	VM	Azure Kubernetes Service (AKS)	Azure Blobs	Instancias
			Azure Container Instances (ACI)	Azure Disks	
			Azure Container Registry	Azure Files	

Fuente: Autor del trabajo.

En la Tabla 3 se presenta la cotización realizada a través de las herramientas Calculadora de Precios, de Azure, AWS y GPC. Se incluye también información respecto a los créditos gratuitos y para estudiantes que ofrece cada uno. Al realizar la comparación de costos entre instancias similares, la diferencia en precios no es realmente significativa, pues aunque, por ejemplo, Amazon tiene un costo superior, también incluye características adicionales como un mayor espacio de almacenamiento. Sin embargo, se observa que Azure, además de ser el más económico, incluye créditos gratuitos, tanto para miembros de la comunidad educativa como para el público en general, obteniendo una posible ventaja en este aspecto.

Tabla 3. Comparación de precios de AWS, GPC y Azure usando su herramienta Calculadora de Precios.

Comparación de precios		AMAZON	GPC	AZURE
Calculadora de precios	Servicio	EC2	Compute Engine	VM
	Tipo de instancia	t3a.xlarge	n1-standard-4	B4MS
	Familia	Propósito general	Propósito general	Máquinas virtuales ampliables
	Sistema Operativo	Linux	Ubuntu	Linux - Ubuntu
	Núm. de instancias.	1	1	1
	vCPUs	4	4	4
	RAM	16 GiB	15 GB	16 GB
	Almacenameinto	EBS, 100GB SSD	375 GB Local SSD	32 GB Almacenamiento temporal
	Tiempo	1 mes, uso constante	720 horas / mes	Mensual
	Fracturación	*On-Demand Instances*	*Regular, None commited usage*	*Pay as you go*
	Total	**145.49**	**141.04**	**121.38**
Prueba gratuita	Tipo de instancia	t2.micro	N/A	B1S
	Tiempo	750 horas	N/A	750 horas
	Créditos gratuitos	0	300 USD	200 USD
Créditos educativos		100 USD	No disponible	100 USD

Fuente: Amazon Web Services, 2019a; Google Cloud, 2019; Microsoft Azure, 2019.

En la Tabla 4 se muestran los resultados de la revisión de antecedentes, en la que se destaca que los servicios de AWS son dominantes en la implementación de flujos bioinformáticos en la nube.

Tabla 4. Resumen de la revisión presentada en los antecedentes.

Nombre del pipeline	Tipo de variante	Alineador implementado	Otras herramientas usadas	CLOUD	PLATAFORMA	TIPO DE DATOS
SIMPLEX	SNPs - indels	BWA	GATK, ANNOVAR	AWS-EC2.	Java EE	WES
RAINBOW	SNPs	Bowtie	PICARD, SOAPsnp	AWS-EC2, s3	N/A	WGS
GenomeKey	SNPs - indels	BWA	GATK	AWS-EC2, s3	COSMOS	WGS
MERCURY	INDELS	BWA	GATK,Atlas-SNP, Atlas-Indel,Cassandra	AWS-EC2, S3	DNAnexus	WGS, WES
Stormseq	SNPS and indels	BWA, BWA-MEM y SNAP.	GATK, Samtools, VEP	AWS-EC2, s3	N/A	WGS, WES

Fuente: Autor del trabajo. Basado en Fischer et al., 2012; Karczewski et al., 2014; Reid et al., 2014; Souilmi et al., 2015; Zhao et al., 2013.

Aunque en los estudios no se especifican comparaciones que lleven a la selección de AWS por encima de las demás ofertas en el mercado, en alguno de ellos se destacan características importantes de este, como su capacidad de escalamiento y paralelización de las tareas con gran facilidad de acuerdo con las necesidades (Zhao et al., 2013), y su seguridad al encapsular toda el sistema del pipeline dentro de un mismo usuario con acceso único a la información (Karczewski et al., 2014).

A continuación se presenta la evaluación final de los proveedores, Tabla 5, donde se califica con una escala de 1 a 5, siendo 5 el valor máximo. Se le otorga una ponderación mayor a la escalabilidad que al costo como tal, debido a que los requerimientos de un flujo bioinformático varían notablemente entre diferentes proyectos según el tamaño de los

datos, y se espera que esta o implementaciones similares puedan ser aplicadas a genomas completos y en cohortes mayores.

Tabla 5. Resultado de la evaluación de los proveedores.

Criterio	%	AZURE		AWS		GPC	
Costo	**20**	5	1	3	0.6	4	0.8
Administración de la infraestructura	**10**	5	0.5	5	0.5	5	0.5
Facilidad de implementación	**20**	3	0.6	4	0.8	5	1
Experiencias en bioinformática	**20**	2	0.4	5	1	3	0.6
Seguridad	**10**	5	0.5	5	0.5	5	0.5
Escalabilidad	**20**	4	0.8	5	1	4	0.8
Total	**100**		**3.8**		**4.4**		**4.2**

Fuente: Autor del trabajo.

A pesar de ser la seguridad de vital importancia y un tema de debate para el manejo de este tipo de datos en un ambiente de nube, en este caso se le da una ponderación baja porque gracias a la gran similitud ofrecida por los tres proveedores en esta materia, no será un factor determinante a la hora de seleccionar. Todos ellos cumplen con estándares homologados en este aspecto (Höfer & Karagiannis, 2011) y con las leyes de protección para los datos clínicos. De manera similar ocurre con la administración de la infraestructura, los tres cuentan con diversas opciones para monitorear los recursos, como el CLI, la consola y una aplicación móvil.

Por otra parte, la facilidad de implementación fue de gran importancia en la selección porque no se tenía experiencia previa con la computación en la nube y se tenía poco tiempo para el desarrollo.

En general, se observa que no hay diferencias significativas en el área de Administración de infraestructura y Seguridad, es decir, los vértices de la Figura 10 convergen para ellos. En los criterios de mayor ponderación: Escalabilidad, Experiencias en bioinformática y Facilidad de implementación, se advierte que AWS está más al exterior en los dos primeros y sólo es superado por GPC en el último.

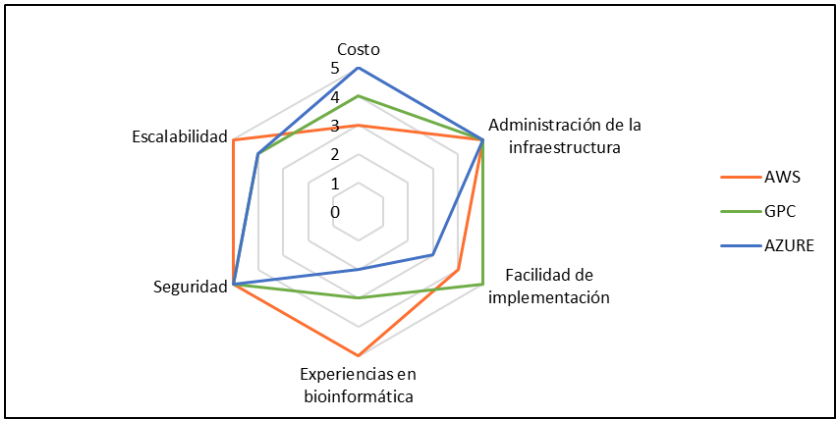

Figura 10. Diagrama de araña como evidencia del resultado de la evaluación comparativa entre los proveedores.

De acuerdo con la información recopilada y la evaluación realizada, se selecciona Amazon como el proveedor del servicio de computación en la nube para el desarrollo del presente trabajo.

3.2 DISEÑO DEL FLUJO DE TRABAJO

La Figura 11 presenta el esquema del flujo de trabajo diseñado. Se inicia con los datos en bruto (raw data) en formato FASTQ porque ha sido el más utilizado para el intercambio de la información de secuencias en los últimos años y el genoma de referencia en el formato que comúnmente se presenta, FASTA, que solo difiere del anterior por la presencia de una calificación numérica para cada nucleótido (Cock, Fields, Goto, Heuer, & Rice, 2010).

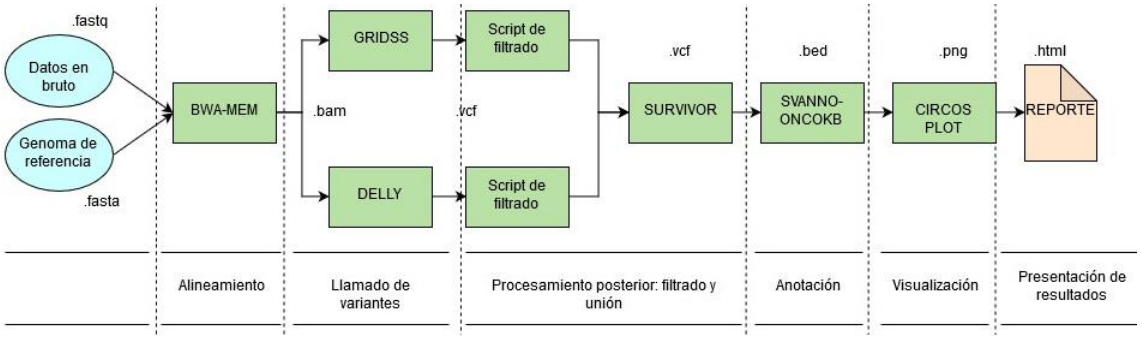

Figura 11. Diseño esquemático del flujo de trabajo implementado.

A partir de los archivos de entrada se empieza con el alineamiento de las lecturas cortas desordenadas para obtener archivos binarios de menor tamaño tipo BAM y se continua con el que podría considerarse el paso más determinante: el llamado de variantes. Es allí donde se identifican con precisión las diferencias y variaciones entre la muestra y el genoma de referencia. Se realiza en dos aplicaciones diferentes, porque los llamados suelen arrojar resultados distintos, y una de las mejores formas de mejorar la calidad de la detección, es realizando este proceso múltiples herramientas para abarcar más variantes. Los resultados de este paso se encuentran en archivos VCF.

Se lleva a cabo un filtrado previo a la unión de variantes, porque lo que se busca con este flujo es ser lo más específico posible, y mostrar los resultados relevantes, más que una gran cantidad de ellos. Se realiza luego la unión de los dos archivos VCF filtrados en uno solo. Y finalmente se procede con la anotación y la generación de gráficas que ilustren los resultados.

A continuación, Tabla 6, se relaciona cada uno de los pasos presentados en la parte inferior del esquemático con las herramientas y aplicaciones usadas para su ejecución, que en la Figura 11 se presentan en color verde. Para los pasos de alineamiento, llamado de variantes y unión de estas, se utilizaron aplicaciones de código abierto que fueron seleccionadas por sus buenas referencias y facilidad de implementación, según los criterios expuestos en la metodología.

Tabla 6. Aplicaciones utilizadas, con su respectiva fuente y proceso al que corresponden.

Proceso	Herramienta - aplicación	Docker Hub	Código	Publicación asociada
Alineamiento	**BWA (Burrows-Wheeler Alignment)**	leukgen/docker-pcapcore	https://github.com/lh3/bwa	Fast and accurate short read alignment with Burrows–Wheeler transform otros (H. Li & Durbin, 2009).
Llamado de variantes	**GRIDSS (Genome Rearrangement IDentification Software Suite)**	papaemmelab/docker-gridss	https://github.com/PapenfussLab/gridss	GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly (Cameron et al., 2017)
Llamado de variantes	**DELLY**	dellytools/delly	https://github.com/dellytools/delly	DELLY: structural variant discovery by integrated paired-end and split-read analysis (Rausch et al., 2012)
Filtrado	**Script**	danielrbroad/pysamdocker	https://github.com/danielavarelat/cli_apps_isabl/blob/master/myapps/myapps/apps/filtering/filtering.py	No aplica
Unión	**SURVIVOR**	No aplica	https://github.com/fritzsedlazeck/SURVIVOR	Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast (Jeffares et al., 2017)
Anotación	**svanno - OncoKB**	danielrbroad/pysamdocker	https://github.com/danielavarelat/svanno	No aplica
Visualización	**Circos plot - Scripts**	papaemmelab/toil_circosigv	https://github.com/danielavarelat/circosplot	No aplica
Presentación de resultados	**Plantilla - html**	No aplica	No aplica	No aplica

Fuente: Autor del trabajo.

Para la anotación se modificó el código fuente de svanno, debido a que este paquete fue desarrollado específicamente para su aplicación en un proyecto del laboratorio Papaemmanuil (Elli Papaemmanuil Lab, 2019) y hacía uso de atributos de las variantes en el formato VCF que no estaban presentes en los resultados de este trabajo. Como se quería hacer modificaciones menores del código sin afectar el proyecto original, se hizo un *fork* del repositorio original. El nuevo código se puede encontrar en el Github, como danielavarelat/svanno.

Para los demás pasos del proceso, es decir, filtrado y generación de circos plot, se desarrollaron scripts propios en Python y R, respectivamente. Se filtró únicamente teniendo en cuenta el filtro PASS de los archivos VCF, que se encuentra en el script filtering.py del repositorio en Github: danielavarelat/cli_apps_isabl. Los scripts para crear las gráficas están disponibles en danielavarelat/circosplot.

El único paso que se realiza localmente es la generación de los circos plot y del reporte, porque la nube es ideal para procesar y monitorear, pero con la experimentación se confirma que hay ciertos procesos que se facilitan si se tienen las herramientas y ambientes de desarrollo ya configurados.

3.3 IMPLEMENTACIÓN

En la Tabla 7 se presenta un resumen de los servicios ofrecidos por AWS que fueron utilizados finalmente como parte de esta implementación, y una breve definición de ellos según la documentación propia de Amazon.

Para el análisis los resultados de la implementación y las aplicaciones específicas de cada servicio, este se dividió en dos etapas.

Tabla 7. Servicios de AWS utilizados.

Servicio	Breve descripción
ECS	"Servicio de administración de contenedores altamente escalable y rápido que facilita la tarea de ejecutar, detener y administrar contenedores de Docker en un clúster" (Amazon Web Services, 2019b).
EC2	"Capacidad de computación escalable en la nube" (Amazon Web Services, 2019a).
ECR	"Registro de contenedores de Docker completamente administrado que facilita a los desarrolladores las tareas de almacenamiento, administración e implementación de imágenes de contenedores de Docker" (Amazon Web Services, 2019e).
IAM	"Administrar el acceso a los servicios y recursos de AWS de manera segura" (Amazon Web Services, 2019c).
EFS	"Suministra un sistema de archivos NFS simple, escalable, elástico y totalmente administrado para utilizar con los servicios en la nube de AWS y los recursos locales" (Amazon Web Services, 2019g).
CloudWatch	"Datos e información procesable para monitorizar sus aplicaciones, responder a cambios de rendimiento, optimizar el uso y lograr una vista unificada del estado " (Amazon Web Services, 2019d).

Fuente: Autor del trabajo.

3.3.1 Despliegue de Isabl en AWS

- **ECR**: para acceder a los contenedores desde los servicios de AWS, es necesario que estén registrados en ECR o disponibles en Docker Hub. Como Isabl es una herramienta en proceso de publicación, se registraron en ECR para su uso privado, como se ve en la Figura 12.

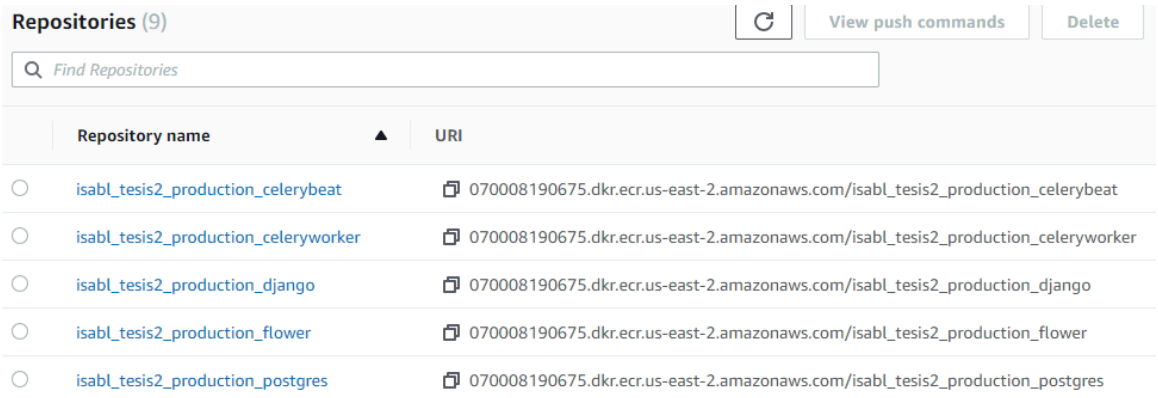

Figura 12. Contenedores que componen Isabl registrados en ECR (Amazon Web Services, 2019i).

- **ECS**: inicialmente se hicieron ensayos en un clúster con una CI de tipo t2.micro, que tiene 1 GiB de memoria, por ser la ofrecida en la prueba gratuita de Amazon, pero probó ser insuficiente para el montaje de Isabl. Se crea, entonces, un clúster,
- Figura 13, con una sola CI de tipo t2.medium, con 4 GiB de memoria y 2 vCPU. Dentro de este se lanza una tarea (*task*) para la ejecución de los cincos contenedores Docker que componen Isabl. En la Figura 14 se evidencia el estado

RUNNING de los contenedores tras el lanzamiento de la tarea y la memoria otorgada a cada uno.

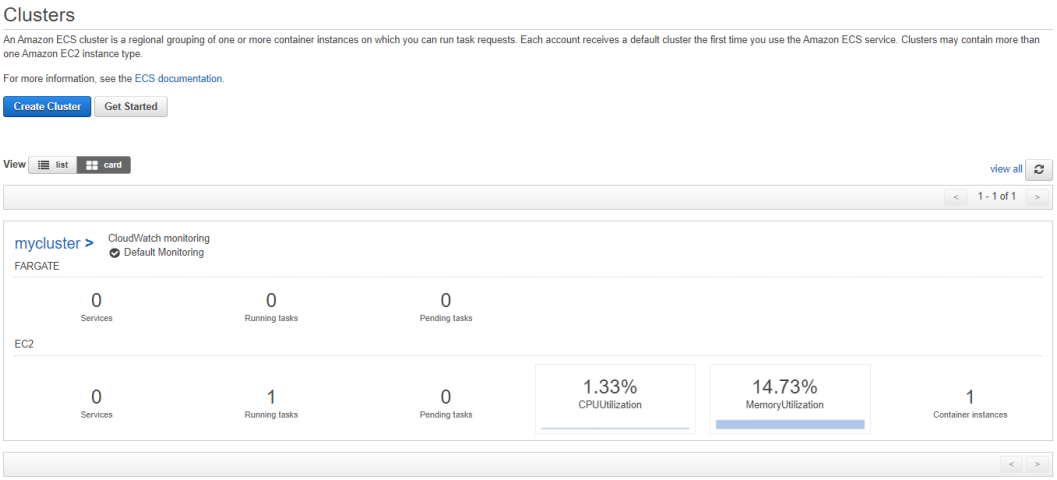

Figura 13. Clúster de instancias de contendores creado con ECS (Amazon Web Services, 2019i).

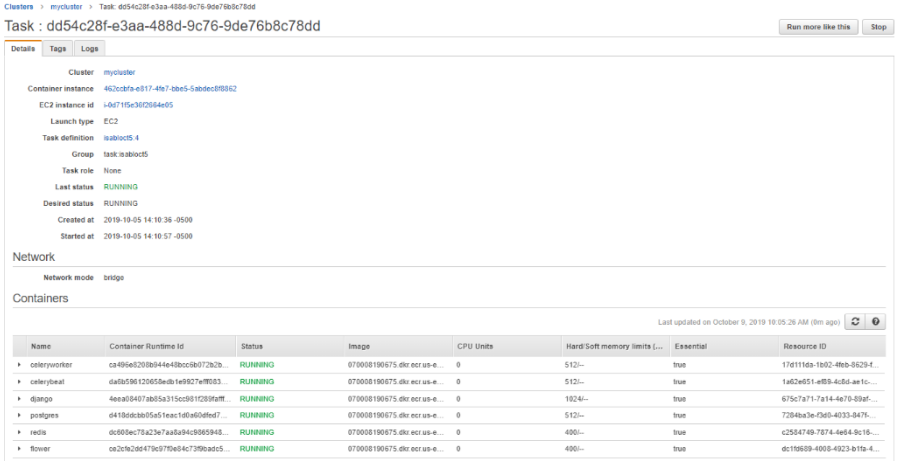

Figura 14. Tarea lanzada con ECS para el despliegue de Isabl (Amazon Web Services, 2019i).

Aunque ECS es compatible con aplicaciones contenerizadas en Docker y Docker Compose, se tuvieron que hacer modificaciones en la configuración de los contenedores y en las especificaciones de red del clúster para permitir la comunicación entre ellos, el acceso a los volúmenes de almacenamiento y la correcta interacción de todos sus componentes.

Se escogió el servicio ECS para la orquestación de contenedores por su gran flexibilidad y capacidad de escalamiento de aplicaciones contenerizadas. Además, ofrece alta disponibilidad y múltiples interfaces de gestión y monitoreo, como la consola y el CLI, facilitando su administración.

En la Figura 15 se muestra el resultado de Isabl funcionando en AWS usando ECS. La IP pública de la CI, fue asociada a un DNS gratuito (http://danielaeia.hopto.org) para mayor facilidad del acceso. Se puede visitar esta página para revisar el proyecto.

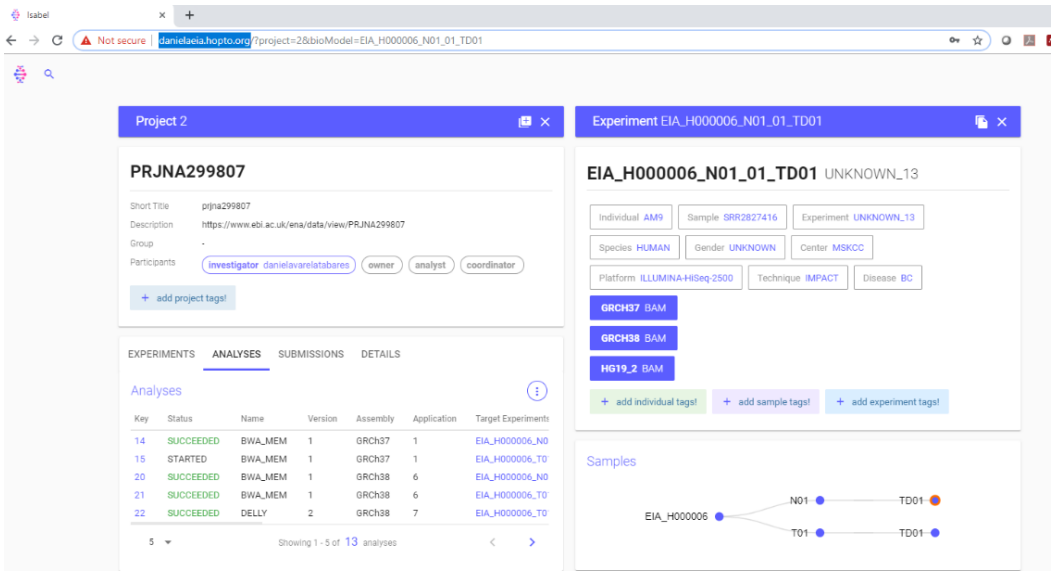

Figura 15. Isabl desplegado en AWS (Amazon Web Services, 2019i).

3.3.2 Ejecución de aplicaciones en la instancia

- **EC2 y CloudWatch**: Amazon EC2 ofrece tres tipos de instancias: bajo demanda, instancias reservadas e instancias de spot. Para esta implementación, aunque sean las más costosas, se escogen las instancias bajo demanda (*On-demand Instances*) porque en estas se cobran únicamente por los segundos en las que se encuentre en estado “RUNNING” y son ideales para aplicaciones de corto tiempo que no serán interrumpidas (Amazon Web Services, 2019k).

  La instancia EC2 se utiliza para la ejecución de las aplicaciones que son lanzadas a través de la línea de comandos de Isabl, aunque también se incluya dentro del servicio EC2 la instancia de contenedores lanzada a través del servicio ECS, explicada anteriormente. Con base en los requerimientos mencionados en la metodología, se lanza una instancia de tipo t2.2xlarge, con 32 GiB de RAM y 8 vCPU, desde la consola de AWS. Se hicieron las instalaciones requeridas como Docker, python, CLI de Isabl y demás repositorios. Se monta el sistema de archivos EFS y se instala el agente CloudWatch para su monitoreo de forma continua. En la Figura 16 se observan las dos instancias, siendo “*processing*" la EC2.

Name	Instance ID	Instance Type	Availability Zone	Instance State
processing	i-04fdfab74bb632fb1	t2.2xlarge	us-east-2b	running
ECS Instance - amazon-ecs-cli-setup-mycluster	i-0d71f5e36f2664e05	t2.medium	us-east-2b	running

Figura 16. Instancias utilizadas para la implementación (Amazon Web Services, 2019i).

  En esta nueva instancia, todos los programas corrieron adecuadamente. Se reportan los resultados de las métricas recopiladas por el agente CloudWatch, tomadas en la instancia durante la ejecución de los diferentes programas. Se hizo uno por uno de forma secuencial para poder evidenciar de manera clara cada desempeño.

Desde la Figura 17 hasta la Figura 20, se muestra en naranja la curva que representa el porcentaje de memoria utilizado, cuyos valores corresponden al eje izquierdo, y en color azul se presenta la información correspondiente al porcentaje de utilización de disco duro en todo momento, eje derecho. Todas las gráficas fueron tomadas como el promedio punto a punto

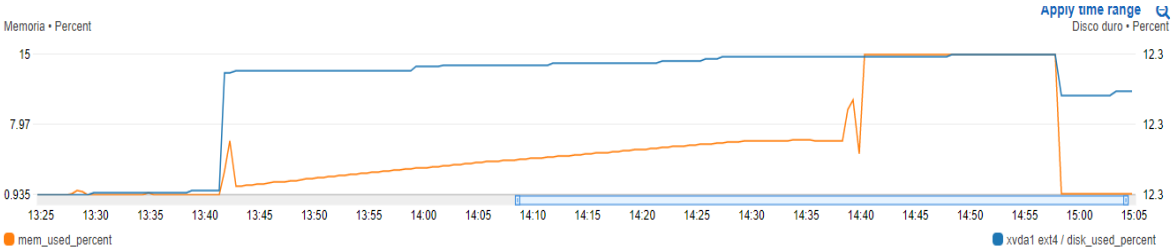

Figura 17. Métricas de memoria y disco duro para el indexado (Amazon Web Services, 2019i).

El proceso de indexado del genoma fue el de mayor duración, con un tiempo de ejecución de 1 hora y 20 minutos, sin embargo, no fue el de mayor consumo de memoria. En la Figura 17, para el uso de memoria, se aprecia un patrón de dos picos repetido, que se debe al alineamiento de la muestra normal y tumor respectivamente, porque fueron lanzadas como una misma tarea. El tiempo de ejecución total fue aproximadamente 40 minutos.

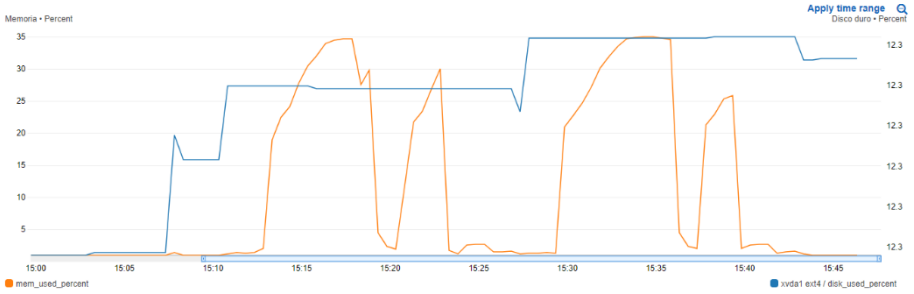

Figura 18. Métricas de memoria y disco duro para BWA (Amazon Web Services, 2019i).

Las dos aplicaciones utilizadas para el llamado de variantes fueron ejecutadas utilizando la muestra tumor y normal dentro del mismo comando, por eso se presenta una única operación. Como se esperaba, DELLY fue el llamador de variantes que menos demanda de recursos presentó y también el más rápido de los procesos, con un tiempo de ejecución de apenas 10 minutos y un uso de memoria menor al 5%.

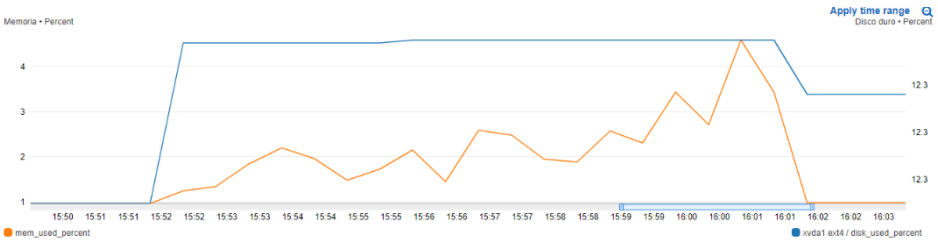

Figura 19. Métricas de memoria y disco duro para DELLY (Amazon Web Services, 2019i).

Durante la ejecución de GRIDSS el proceso falló, debido a falta de espacio de almacenamiento, por lo que fue necesario escalar el volumen EBS adherido a la instancia a 64 GiB, el cual originalmente era de 8 GiB. Como se muestra en la Figura 18, fue la operación más demandante y genera una gran cantidad de archivos intermedios que consumen el almacenamiento local de la instancia. Es la única de las aplicaciones que supera el 50% de memoria y que presenta picos pronunciados por encima del 13% para el uso del disco duro. Este comportamiento concuerda con lo esperado según los requerimientos que guiaron a la selección de la instancia.

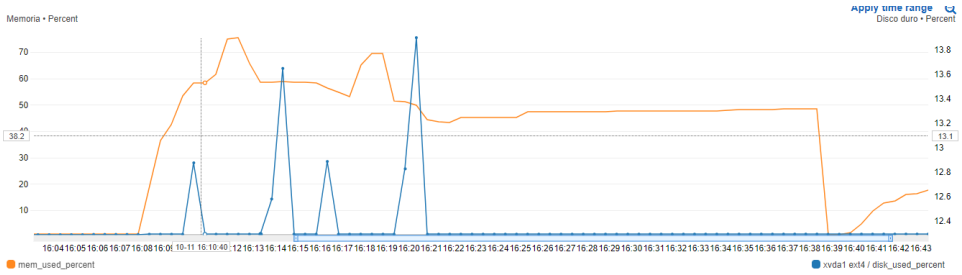

Figura 20. Métricas de memoria y disco duro para GRIDSS (Amazon Web Services, 2019i).

Los procesos finales del flujo, a partir de la obtención del archivo VCF, fueron los menos demandantes de recursos, tal como se esperaba. En la Figura 21 se evidencian algunos picos que no superan el 2% de memoria y un tiempo de ejecución muy reducido.

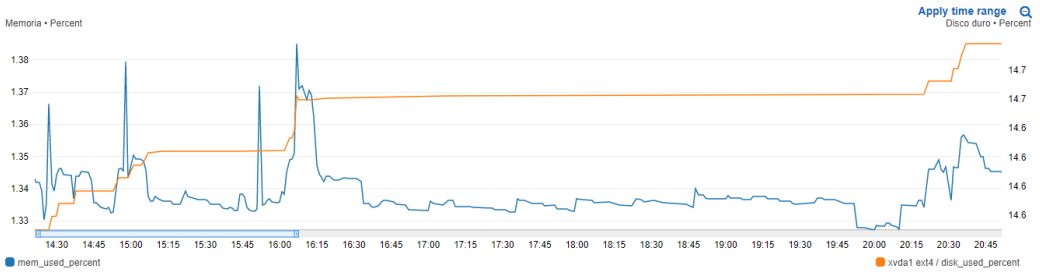

Figura 21. Métricas de memoria y disco duro para filtrado, unión y anotación (Amazon Web Services, 2019i).

- **EFS**: después de toda la implementación, el sistema de archivos creado con EFS alcanzó un tamaño de 53.8 GiB, Figura 22. Esto incluye el genoma de referencia y sus archivos de indexación asociados (20 GiB, por las dos versiones), datos de entrada para ambos estudios (aproximadamente 15 GiB), los resultados de cada uno de los pasos, las referencias utilizadas para la anotación, las instalaciones para la unión de variantes y el repositorio que contiene el CLI y las aplicaciones, entre otros.

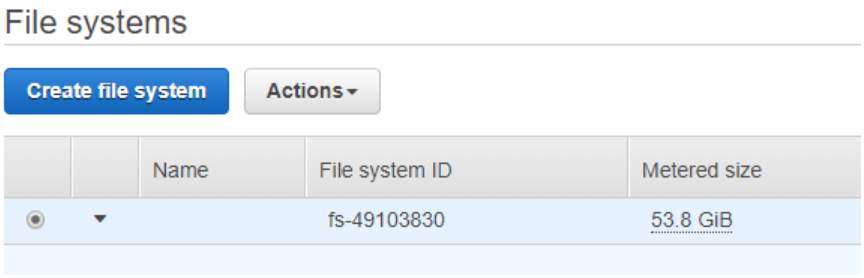

Figura 22. Sistema de archivos creado con EFS (Amazon Web Services, 2019i).

Utilizar EFS fue ideal para facilitar el escalamiento. El sistema de archivos no depende de la instancia y fue fácilmente montado en las instancias EC2. Además, permite a los comandos básicos de Isabl que fueron diseñados para explorar recursivamente los directorios, funcionar de manera natural sin necesidad de cambios adicionales.

- **IAM**: en total se crearon 14 roles, Figura 23. Además, se crea un usuario (*Administrator*), y un único grupo para él (thesis_daniela). Estos roles permitieron controlar la autenticación de identidades, credenciales y el acceso a los diferentes servicios y recursos de forma segura. En sus políticas se definen las acciones que se pueden ejecutar desde la cuenta asociada.

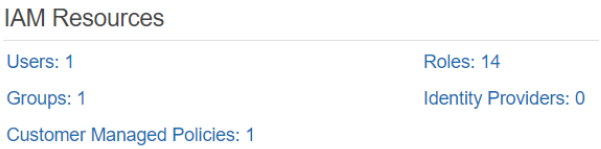

Figura 23. Utilización del servicio IAM (Amazon Web Services, 2019i).

3.4 EVALUACIÓN

3.4.1 Validación de variantes

Se puede acceder a los reportes disponibles libremente en S3, usando los enlaces https://isabltest.s3.us-east-2.amazonaws.com/Resultados/rep807.html, donde se presentan las variantes encontradas. A continuación se discuten los resultados.

En la Figura 22 se observa el CIRCOS plot donde se muestran las variantes en su respectivo cromosoma. Las anotaciones de genes que se muestran en esta gráfica se hacen a partir de un archivo que se había usado previamente durante el trabajo realizado en el laboratorio Papaemmanuil para la misma aplicación, se puede encontrar en danielavarelat/circosplot.

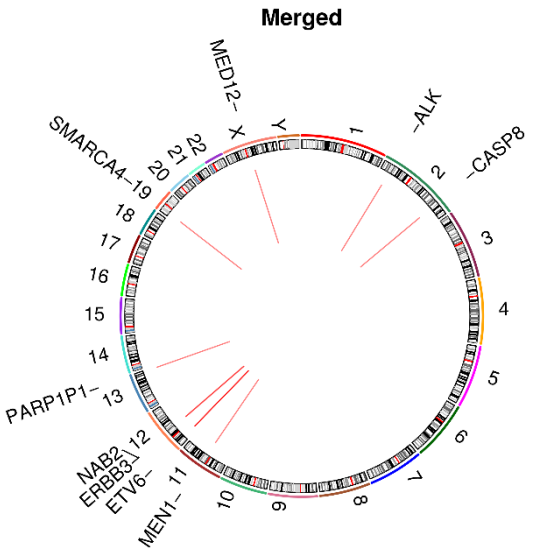

Figura 24. Circos plot resultado de la unión del estudio PRJNA299807.

En la Tabla 8 se encuentra la anotación realizada con svanno de las SVs encontradas después del proceso de unión con SURVIVOR, en el que se dejan únicamente las variantes con máximo 1kb de distancia entre los puntos de quiebre, que sean soportadas por ambos llamadores y de por lo menos 30bp de longitud.

Tabla 8. Anotación de variantes encontradas.

	chr1	chr2	start1	end1	name	type	Annotation
1	11	11	64577391	64577392	INV(chr11:64577392-chr11:64577638)	INV	MEN1(TSG):exon2>>MEN1(TSG):intron1
2	12	12	12014119	12014120	INV(chr12:12014120-chr12:12014351)	INV	ETV6(TSG):intron4>>ETV6(TSG):intron4
3	12	12	12017116	12017117	INV(chr12:12017117-chr12:12017238)	INV	ETV6(TSG):intron4>>ETV6(TSG):intron4
4	12	12	12026414	12026415	INV(chr12:12026415-chr12:12027154)	INV	ETV6(TSG):intron5>>ETV6(TSG):intron5
5	12	12	56493744	56493745	INV(chr12:56493745-chr12:56494031)	INV	ERBB3(ONC):exon25>>ERBB3(ONC):intron26
6	12	12	57487031	57487032	INV(chr12:57487032-chr12:57487098)	INV	NAB2>>NAB2
7	13	13	111590550	111590551	INV(chr13:111590551-chr13:111590743)	INV	Intergenic>>Intergenic
8	19	19	11097666	11097667	INV(chr19:11097667-chr19:11099267)	INV	SMARCA4(TSG):exon5>>SMARCA4(TSG):intron6
9	2	2	29447630	29447631	INV(chr2:29447631-chr2:29447917)	INV	ALK(ONC):intron19>>ALK(ONC):intron19
10	2	2	202146647	202146648	DEL(chr2:202146648-chr2:202149441)	DEL	CASP8(TSG):intron8>>CASP8(TSG):intron8

Fuente: Autor del trabajo.

Comparando la anotación del CIRCOS plot respecto a la mostrada en la tabla anterior, sólo se diferencian en la variante del cromosoma 13 que no se encuentra reportada en OncoKB, pero sí se encontraron reportes de la relación entre la regulación genética de PARP-1 y la carcinogénesis (Lockett, Snowhite, & Hu, 2005). Todas las variantes, excepto la deleción encontrada en el cromosoma 12, son inversiones, por lo que su representación es igual en el CIRCOS plot.

En la anotación se presenta información relacionada con ambos puntos de quiebre y si corresponden a intrones o exones, genes oncogénicos (ONC) o genes supresores tumorales (TSG), según los genes reportados en OncoKB. De todas las variantes

encontradas solo una no presenta anotación, como se mencionó anteriormente. Para los demás, se analizan los hallazgos.

El estudio de Honda et al. (2004) reporta que la alteración del gen supresor de tumor, MEN 1, puede ser responsable del desarrollo de cáncer de mama. El gen ETV 6 se encuentra reportado dentro del estudio base (Geyer et al., 2018) como parte del repertorio de alteraciones genéticas somáticas en adenomioepitelioma de mama, específicamente como parte de las alteraciones en el número de copias.

El gen oncogénico ERBB3 también es parte del repertorio de alteraciones del estudio (Geyer et al., 2018), y ha sido asociado en otros estudios con la expresión del cáncer mamario y la resistencia a la terapia (Lemoine et al., 1992; Stern, 2008).

Para los demás genes: NAB2, SMARCA4, ALK y CASP8 también se encontró evidencia que los vincula con el cáncer de mama. Los dos primeros, como reguladores de expresión de otros genes importantes en el desarrollo de la enfermedad (Kumbrink & Kirsch, 2012; Schramedei et al., 2011), ALK asociado principalmente al cáncer de mama inflamatorio (Tuma, 2012), y CASP8, implicado en la apoptosis de las células cancerígenas, se ha relacionado con el riesgo de padecer cáncer de mama (Cox et al., 2007).

La contrastación de los resultados con respecto a estudios anteriores demuestra la efectividad del flujo de trabajo implementado, al poder verificar la relación de las variantes estructurales encontradas en las muestras, con la enfermedad. Cabe resaltar, que el objetivo de este trabajo no es calificar la precisión de detección de los algoritmos ni establecer criterios para su ejecución y selección.

Es de gran importancia resaltar que el estudio de Geyer et al. (2018), se utiliza únicamente con el propósito de tener un contexto para los datos analizados y poder relacionarlos con el tipo de cáncer del paciente. Sin embargo, no resultaría adecuado realizar una comparación detallada de los resultados porque los objetivos, técnicas y metodologías empleadas divergen. Se debe hacer una validación rigurosa en caso de publicación o implementación del flujo en ambientes clínicos.

3.4.2 Análisis del costo de implementación

Se presentan los resultados tomados con la herramienta *Cost Explorer* de AWS, donde se detallan los gastos asociados a los diferentes servicios. En la Figura 24, se muestra el costo diario variable, evidenciando un modelo de facturación por demanda. En el eje vertical, los costos se presentan en dólares.

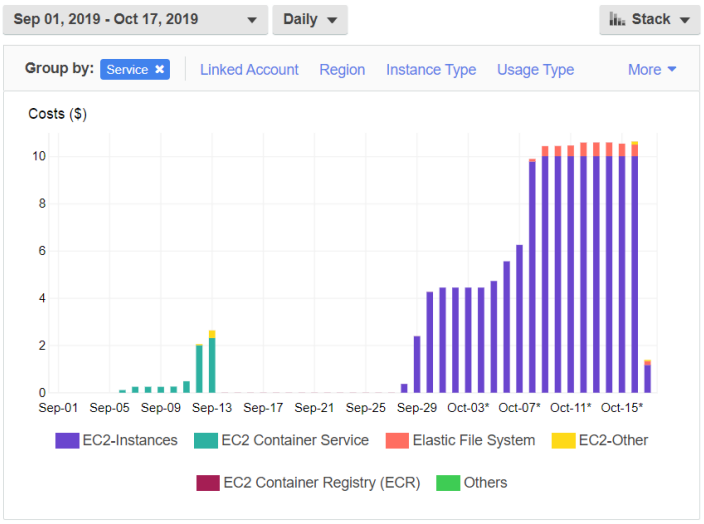

Figura 25. Facturación por servicio por día (Amazon Web Services, 2019i).

En los primeros días, entre el 05 y 13 de septiembre, se observan los costos asociados a una etapa inicial de experimentación con la plataforma y el servicio ECS. Allí se encuentra un poco de color amarillo en las barras, identificado como EC2-Other, y corresponde específicamente a las pruebas iniciales realizadas con Fargate. Después sigue un periodo de inactividad debido a adecuaciones y desarrollos locales, que sirvieron para realizar la implementación definitiva, parte final de la gráfica. Entre los días 01 y 07 de octubre se trabajó con una instancia t2.xlarge, que luego fue escalada a una t2.2xlarge, indicada por el aumento del costo en la figura.

A partir de ese momento, se realizaron las pruebas con los datos reales, los cuales fueron almacenados en el sistema de archivos EFS, color naranja, cuyo tamaño fue aumentando paulatinamente con la generación de nuevos archivos. La última barra de menor altura corresponde al momento en el que se deja de usar la instancia de procesamiento.

En la Tabla 9 se calcula el costo efectivo de la implementación, de acuerdo con las aproximaciones descritas en la metodología. El costo de la instancia de procesamiento supera en gran medida los demás costos, en ella se llevan a cabo los procesos realmente demandantes de recursos, por lo que se selecciona como el punto de análisis y comparación con otras cotizaciones. La instancia de contenedores donde se encuentra Isabl, tiene un costo aproximado de 1.1 dólares por día, teniendo en cuenta que su uso es ininterrumpido.

Tabla 9. Costo real de la implementación (Todos los costos se presentan en dólares)

Servicio	Costo	Unidad de cobro	Cantidad	Costo total
t2.2xlarge	0.3712	USD/h	96	35.6352
t2.medium	0.0464	USD/h	120	5.568
EBS	0.1	USD/GiB-Mes	10	1
EFS	0.3	USD/GiB-Mes	10	3
ECR	0.1	USD/GiB-Mes	0.6	0.06
Total				45.2632

Fuente: Autor del trabajo.

En los resultados anteriores es importante tener en cuenta que los precios corresponden a la región US East (Ohio), identificada como us-east-2, seleccionada por defecto, y que para este caso se usaron instancias bajo demanda AWS, como se expuso anteriormente.

No se tuvieron en cuenta las promociones ofrecidas por la prueba gratuita de AWS, ni los costos asociados al agente CloudWatch, porque su cobro empieza a partir de la décima métrica y no es necesario más que 10 para este caso. ECS no presenta cargos adicionales, únicamente cobra por los recursos EC2 que se utilicen.

Según la tasa de cambio representativa del mercado para el día 19 de octubre ($3,465.35COP), el costo final equivale a $156,853 COP (Dolar Web, 2019). Este valor permite hacerse una idea de la inversión necesaria para implementar flujos bioinformáticos similares, e incluso sirve como punto de partida para un posible escalamiento de los recursos.

Para la comparación con BIOS, de la cotización recibida, se toma la instancia denominada mínima, Figura 26. Aunque durante la implementación se usó una instancia t2.2xlarge, con objeto de la comparación se eligió una instancia t3a.2xlarge, Figura 27, por ser la que presenta especificaciones técnicas semejantes a la instancia mínima de BIOS, y por ser las más cercanas a la infraestructura realmente usada.

INSTANCIA		VALOR MES 24/7
Instancia mínima		$1.108.356
cores	4	
Ram GB	32	
HDD GB	150	

Figura 26. Cotización de BIOS (BIOS, 2019).

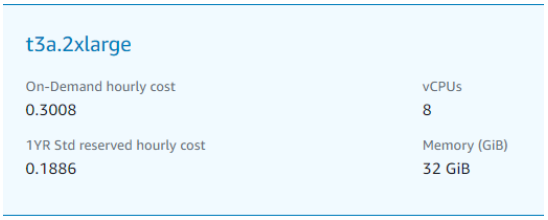

Figura 27. Especificaciones técnicas de instancia t3a.2xlarge de AWS (Amazon Web Services, 2019h).

Tabla 10. Costos mensuales por servicio AWS.

Servicio	**Dólares por mes**
Instancia EC2 (t3a.2xlarge)	219.58
EBS (64 GiB)	6.4
EFS (86 GiB)	25.8
Total por mes	**251.78**

Fuente: Autor del trabajo.

En el caso de la instancia EC2, se hizo la estimación con un uso ininterrumpido de la misma durante un mes. Para el almacenamiento, se repartieron los 150 GiB ofrecidos por BIOS, los cuales son unificados, en los dos servicios de AWS usados que responden a la misma necesidad, EBS y EFS, Tabla 10. El costo mensual en pesos colombianos para esta estimación con AWS es de $872,506 COP, mientras el valor mensual para BIOS es de $1,108,356 COP.

Para este caso de implementación, la comparación muestra que resulta más costoso utilizar una infraestructura con las características anteriores en un centro de computación de alto rendimiento que por medio de AWS. El modelo bajo demanda que se maneja en la computación en la nube hace que sea más flexible en cuanto a los recursos y se puedan optimizar de acuerdo con las necesidades particulares de los clientes.

Cabe resaltar que implementaciones bioinformáticas, como esta o más complejas, pueden ser implementadas en AWS por costos menores a los expuestos. Para esta se utilizaron instancias bajo demanda, principalmente por el tiempo de entrega, pero el ahorro podría ser mucho mayor si se utilizan instancias Spot, o incluso instancias reservadas, por ejemplo en el caso de centros de investigaciones que tengan una buena planeación y detalle de los requerimientos, y realicen implementaciones a un plazo mayor.

El análisis realizado ilustra también la versatilidad del servicio al posibilitar el escalamiento sin mayor inconveniencia en el momento que se requiera, como se hizo para la instancia EC2 de procesamiento cuando cambiaron los requerimientos.

4. CONCLUSIONES Y CONSIDERACIONES FINALES

En el desarrollo de este proyecto se diseñó e implementó con éxito un flujo de trabajo bioinformático en la nube y se probó su funcionamiento utilizando datos oncogénicos provenientes de una base de datos pública. Isabl demostró ser una herramienta útil y eficaz para el despliegue sistemático de las operaciones y con gran potencial para ser implementada en diferentes ambientes. Aunque en este caso se usaron pocos datos y un único paciente, podría extenderse a grandes cohortes e innumerables aplicaciones desde diversos paradigmas de programación.

Debido al abanico tan extendido de ofertas de computación en la nube que se tiene actualmente en el mercado y la similitud de los servicios que ofrecen las diferentes compañías, se hace necesario establecer previa y claramente las necesidades del proyecto para seleccionar el proveedor que ofrezca une verdadera ventaja competitiva.

La comparación de los servicios particulares, entre proveedores, resulta una tarea dispendiosa porque ofrecen configuraciones diferentes, lo que hace difícil encontrar un punto de comparación equivalente. En este trabajo se quiso hacer una comparación teniendo en cuenta únicamente un subconjunto reducido de los servicios necesarios para este tipo de aplicaciones y que sirva a futuros usuarios como punto de partida, tanto para encontrar fuentes y criterios de evaluación, como para la selección.

La computación en la nube demostró ser una opción viable en términos de costos para esta implementación y posiblemente para similares, en las que se tiene poco capital de inversión, se requiere realizar tareas computacionales, que sería imposible realizar en la máquina local, y se tienen flujos de trabajo establecidos. Dentro de las grandes ventajas que ofrece la computación en la nube se encuentra la de no incurrir en grandes inversiones iniciales de infraestructura física e informática, y en gastos periódicos de mantenimiento y personal. Además, los beneficios que trae el cobro bajo demanda, la gran oferta de servicios amigables que optimizan los recursos y facilitan la programación, monitoreo y despliegue de estos, la escalabilidad y comodidad de uso. Todos estos fueron evidenciados durante el desarrollo de la implementación.

Una de las limitaciones de este trabajo radica en la dificultad que se tiene para acceder a cotizaciones de los centros de computación de alto rendimiento en Colombia, fácilmente entendibles por el usuario, que sean equivalentes con servicios en la nube para hacer una comparación amplia de la oferta, y poder sacar conclusiones más generalizadas sobre la viabilidad de ambos según las necesidades. Podría ser objeto de otro estudio, comparar los costos de un proyecto de investigación a mediano o largo plazo, en un centro de computación de alto rendimiento, respecto a la nube. Según los resultados obtenidos en los diagramas de cobro de AWS, se podría anticipar que para una investigación de mayor duración, donde se deben hacer pruebas y ensayos de forma continua, posiblemente la nube no sea la opción más viable en términos económicos.

En medio del auge de la genómica y de la medicina personalizada o de precisión, el desarrollo de flujos de trabajo, como el presentado, cuyo objetivo principal es convertir los datos de secuenciación en bruto en información clínica relevante para el médico, que guíen el diagnóstico y la selección de tratamiento para enfermedades como el cáncer, cobran

cada vez más importancia. Sin embargo, en países como Colombia, la falta de experticia e inversión en investigación limita la implementación de estas prácticas en el ámbito clínico, principalmente por la imposibilidad de acceder a infraestructuras para el análisis de este tipo de datos. Lo anterior, resalta la importancia de explorar nuevas alternativas como la computación en la nube y acercarse a soluciones para pequeños centros de investigación, que les permitan sacar el mayor provecho de las tecnologías que están revolucionando la medicina internacional.

Los resultados obtenidos con la implementación tienen también un valor adicional por la forma en que se presentan. La generación de reportes con información clínica relevante facilita la interpretación y amplía el público de interés, al incluir desde personas con conocimiento en bioinformática, hasta médicos que únicamente requieran conocer las variantes finales detectadas y el gen al que pertenecen. Este es un factor diferenciador respecto a la mayoría de los estudios similares que se revisaron en el estado del arte.

Es pertinente resaltar que la validación de variantes estructurales encontradas es adecuada y fue suficiente para el alcance de este trabajo, pero para su implementación en una cohorte mayor o su posible aplicación en ambientes investigativos y clínicos, debe hacerse una validación más rigurosa y hacer un seguimiento más detallado de los resultados desde el punto de vista biológico, preferiblemente con un equipo multidisciplinario que los apruebe.

De acuerdo con la experiencia obtenida durante el proceso de realización del presente trabajo, se presentan las siguientes sugerencias para futuros proyectos similares en el área de computación en la nube y bioinformática. Para lo primero, se recomienda hacer un adecuado proceso de levantamiento de requerimientos, especialmente del uso de memoria y almacenamiento, para no incurrir en gastos adicionales o invertir el tiempo en ejecuciones fallidas. Para lo segundo, debido a la inmensa cantidad de aplicaciones que se encuentran para cada paso de los flujos de trabajo bioinformático, se recomienda hacer una buena revisión de la documentación de cada uno y contrastarlo con las necesidades particulares del proyecto.

REFERENCIAS

Afgan, E., Baker, D., Batut, B., van den Beek, M., Bouvier, D., Čech, M., … Blankenberg, D. (2018). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research*, *46*(W1), W537–W544. https://doi.org/10.1093/nar/gky379

Afgan, E., Baker, D., Coraor, N., Chapman, B., Nekrutenko, A., & Taylor, J. (2013). *Galaxy CloudMan: delivering cloud compute clusters*. *11*(Suppl 12), 1–6. Retrieved from papers2://publication/uuid/60694EA3-147A-4C29-A160-43819BCB0940

Amazon Web Services, I. (2017). Announcing Amazon EC2 per second billing. Retrieved October 18, 2019, from https://aws.amazon.com/about-aws/whats-new/2017/10/announcing-amazon-ec2-per-second-billing/

Amazon Web Services, I. (2018). EC2 or AWS Fargate? Retrieved October 15, 2019, from https://containersonaws.com/introduction/ec2-or-aws-fargate/

Amazon Web Services, I. (2019a). ¿Qué es Amazon EC2? - Amazon Elastic Compute Cloud. Retrieved October 15, 2019, from https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html

Amazon Web Services, I. (2019b). ¿Qué es Amazon Elastic Container Service? - Amazon Elastic Container Service. Retrieved October 15, 2019, from https://docs.aws.amazon.com/es_es/AmazonECS/latest/developerguide/Welcome.html

Amazon Web Services, I. (2019c). Administración de identidades | IAM | AWS. Retrieved October 15, 2019, from https://aws.amazon.com/es/iam/

Amazon Web Services, I. (2019d). Amazon CloudWatch: Monitoreo de infraestructuras y aplicaciones. Retrieved October 15, 2019, from https://aws.amazon.com/es/cloudwatch/

Amazon Web Services, I. (2019e). Amazon ECR | Amazon Web Services. Retrieved October 15, 2019, from https://aws.amazon.com/es/ecr/

Amazon Web Services, I. (2019f). Amazon Elastic Block Store (EBS) - Amazon Web Services. Retrieved October 22, 2019, from https://aws.amazon.com/ebs/

Amazon Web Services, I. (2019g). Amazon Elastic File System (EFS) | Almacenamiento de archivos en la nube. Retrieved October 15, 2019, from https://aws.amazon.com/es/efs/

Amazon Web Services, I. (2019h). AWS Pricing Calculator. Retrieved October 15, 2019, from https://calculator.aws/#/

Amazon Web Services, I. (2019i). EC2 Management Console. Retrieved October 21, 2019, from https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Home:

Amazon Web Services, I. (2019j). Instancias de contenedor de Amazon ECS. Retrieved October 15, 2019, from https://docs.aws.amazon.com/es_es/AmazonECS/latest/developerguide/ECS_instances.html

Amazon Web Services, I. (2019k). On-Demand Instances - Amazon Elastic Compute Cloud. Retrieved October 18, 2019, from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html

Amazon Web Services, I. (2019l). What is Amazon S3? - Amazon Simple Storage Service. Retrieved October 22, 2019, from https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

Amazon Web Services, I. (2019m). What is AWS. Retrieved October 22, 2019, from https://aws.amazon.com/what-is-aws/

Angiuoli, S. V., White, J. R., Matalka, M., White, O., & Fricke, W. F. (2011). Resources and Costs for Microbial Sequence Analysis Evaluated Using Virtual Machines and Cloud Computing. *PLoS ONE*, *6*(10), e26624. https://doi.org/10.1371/journal.pone.0026624

Bala, R., Gill, B., Smith, D., & Wright, D. (2019). Gartner Reprint. Retrieved October 11, 2019, from Magic Quadrant for Cloud Infrastructure as a Service, Worldwide website: https://www.gartner.com/doc/reprints?id=1-1CMAPXNO&ct=190709&st=sb

Barabas, J. (n.d.). IaaS PaaS SaaS Cloud Service Models | IBM Cloud. Retrieved April 29, 2019, from https://www.ibm.com/cloud/learn/iaas-paas-saas

Bianchi, V., Ceol, A., Ogier, A. G. E., de Pretis, S., Galeota, E., Kishore, K., … Pelizzola, M. (2016). Integrated Systems for NGS Data Management and Analysis: Open Issues and Available Solutions. *Frontiers in Genetics*, *7*, 75. https://doi.org/10.3389/fgene.2016.00075

BIOS. (2019). Centro de Bioinformática y Biología Computacional de Colombia - BIOS - BIOS. Retrieved October 18, 2019, from http://bios.co/

Broad Institute. (2019). FIRECLOUD. Retrieved October 11, 2019, from https://software.broadinstitute.org/firecloud/

Cameron, D. L., Schröder, J., Penington, J. S., Do, H., Molania, R., Dobrovic, A., … Papenfuss, A. T. (2017). GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. *Genome Research*, *27*(12), 2050–2060. https://doi.org/10.1101/gr.222109.117

Chakravarty, D., Gao, J., Phillips, S. M., Kundra, R., Zhang, H., Wang, J., … Schultz, N. (2017). OncoKB: A Precision Oncology Knowledge Base. *JCO Precision Oncology*, *2017*. https://doi.org/10.1200/PO.17.00011

Chao, K.-M. (2006). *Basic Concepts of DNA, Proteins, Genes and Genomes*. Retrieved from

https://www.csie.ntu.edu.tw/~kmchao/seq06fall/genome.pdf

Clancy, S. (2008). Genetic Mutation. *Nature Education*, *1*. Retrieved from https://www.nature.com/scitable/topicpage/genetic-mutation-441

Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L., & Rice, P. M. (2010). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, *38*(6), 1767–1771. https://doi.org/10.1093/nar/gkp1137

Cortés Martín, A. (2014). Introducción a la Línea de comandos en Windows. Retrieved October 22, 2019, from http://www.it.uc3m.es/java/git-gisc/resources/windows-command-line/windows-command-line.html

Cox, A., Dunning, A. M., Garcia-Closas, M., Balasubramanian, S., Reed, M. W. R., Pooley, K. A., … Easton, D. F. (2007). A common coding variant in CASP8 is associated with breast cancer risk. *Nature Genetics*, *39*(3), 352–358. https://doi.org/10.1038/ng1981

DNANEXUS, I. (2019). DNAnexus. Retrieved October 11, 2019, from https://www.dnanexus.com/

Docker Inc. (2019). What is a Container? | Docker. Retrieved October 22, 2019, from https://www.docker.com/resources/what-container

Dolar Web. (2019). Dolar TRM Hoy para Colombia, Histórico ultimos 30 dias COP $ 3465.35 - Dolar Web. Retrieved October 18, 2019, from https://dolar.wilkinsonpc.com.co/divisas/dolar.html

Elli Papaemmanuil Lab. (2019). papaemmelab/svanno: Annotate SVs. Retrieved October 15, 2019, from https://github.com/papaemmelab/svanno

EMBL-EBI. (n.d.). What is Next-Generation DNA Sequencing | EMBL-EBI Train online. Retrieved April 29, 2019, from https://www.ebi.ac.uk/training/online/course/ebi-next-generation-sequencing-practical-course/what-you-will-learn/what-next-generation-dna-

Encyclopædia Britannica, I. (2019). Base pair| Britannica.com. Retrieved October 22, 2019, from https://www.britannica.com/science/base-pair

Escaramís, G., Docampo, E., & Rabionet, R. (2015). A decade of structural variants: description, history and methods to detect structural variation. *Briefings in Functional Genomics*, *14*(5), 305–314. https://doi.org/10.1093/bfgp/elv014

Fernald, G. H., Capriotti, E., Daneshjou, R., Karczewski, K. J., & Altman, R. B. (2011). Bioinformatics challenges for personalized medicine. *Bioinformatics*, *27*(13), 1741–1748. https://doi.org/10.1093/bioinformatics/btr295

Fischer, M., Snajder, R., Pabinger, S., Dander, A., Schossig, A., Zschocke, J., … Stocker, G. (2012). SIMPLEX: Cloud-enabled pipeline for the comprehensive analysis of exome sequencing data. *PLoS ONE*, *7*(8), 1–8. https://doi.org/10.1371/journal.pone.0041948

FUGA BV. (2019). Cloud Instances: All the compute power you'll ever need - Fuga Cloud. Retrieved October 22, 2019, from https://fuga.cloud/tag/instance/

Gafni, E., Luquette, L. J., Lancaster, A. K., Hawkins, J. B., Jung, J.-Y., Souilmi, Y., … Tonellato, P. J. (2014). COSMOS: Python library for massively parallel workflows. *Bioinformatics*, *30*(20), 2956–2958. https://doi.org/10.1093/bioinformatics/btu385

Geyer, F. C., Li, A., Papanastasiou, A. D., Smith, A., Selenica, P., Burke, K. A., … Reis-Filho, J. S. (2018). Recurrent hotspot mutations in HRAS Q61 and PI3K-AKT pathway genes as drivers of breast adenomyoepitheliomas. *Nature Communications*, *9*(1), 1816. https://doi.org/10.1038/s41467-018-04128-5

Google Cloud. (2019). Calculadora de precios de Google Cloud Platform. Retrieved October 15, 2019, from https://cloud.google.com/products/calculator/?hl=es-419

Griffith, M., Griffith, O. L., Smith, S. M., Ramu, A., Callaway, M. B., Brummett, A. M., … Wilson, R. K. (2015a). Genome Modeling System: A Knowledge Management Platform for Genomics. *PLOS Computational Biology*, *11*(7), e1004274. https://doi.org/10.1371/journal.pcbi.1004274

Griffith, M., Griffith, O. L., Smith, S. M., Ramu, A., Callaway, M. B., Brummett, A. M., … Wilson, R. K. (2015b). Genome Modeling System: A Knowledge Management Platform for Genomics. *PLOS Computational Biology*, *11*(7), e1004274. https://doi.org/10.1371/journal.pcbi.1004274

Grubka, S., & Jacobs, H. (2014). Understanding the Human Genome Project. *Phi Delta Kappan*, *86*(4), i–i. https://doi.org/10.1177/003172170408600401

Guan, P., & Sungab, W.-K. (2016). Structural variation detection using next-generation sequencing data: A comparative technical review. *Methods*, *102*, 36–49. https://doi.org/10.1016/J.YMETH.2016.01.020

Guru99. (2019). Agile Model & Methodology: Guide for Developers and Testers. Retrieved October 13, 2019, from https://www.guru99.com/agile-scrum-extreme-testing.html

Hayes, M. (2019). Computational Analysis of Structural Variation in Cancer Genomes. In *Methods in molecular biology (Clifton, N.J.)* (Vol. 1878, pp. 65–83). https://doi.org/10.1007/978-1-4939-8868-6_3

Hewlett Packard Enterprise Development LP. (2019). ¿Qué es la secuenciación de nueva generación? - Definiciones de TI empresarial | HPE España. Retrieved April 29, 2019, from https://www.hpe.com/es/es/what-is/next-gen-sequencing.html

Hille, M., & Baum, M. (2018). *CLOUD COMPUTING VENDOR & SERVICE PROVIDER COMPARISON*. Retrieved from https://d1.awsstatic.com/analyst-reports/Report_CVU_CC_AWS_ENGL_final.pdf

Hille, M., Klemm, D., & Lemmermann, L. (2017). *CLOUD COMPUTING VENDOR &*

SERVICE PROVIDER COMPARISON. Retrieved from https://www.reply.com/Documents/Crisp_Vendor_Universe_Cloud Computing_250118_REPLY_englischeVersion_FINAL.pdf

Höfer, C. N., & Karagiannis, G. (2011). Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*, *2*(2), 81–94. https://doi.org/10.1007/s13174-011-0027-x

Illumina, I. (2019a). DNA Sequencing | Understanding the genetic code. Retrieved October 11, 2019, from https://www.illumina.com/techniques/sequencing/dna-sequencing.html

Illumina, I. (2019b). Paired-End vs. Single-Read Sequencing Technology. Retrieved October 22, 2019, from https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/paired-end-vs-single-read.html

Illumina, I. (2019c). Targeted Resequencing | Focused investigation of key genes. Retrieved October 14, 2019, from https://www.illumina.com/techniques/sequencing/dna-sequencing/targeted-resequencing.html

InfoSoft Global Private Limited. (2019). Radar Chart - A Complete Guide | FusionCharts. Retrieved October 11, 2019, from https://www.fusioncharts.com/resources/chart-primers/radar-chart

Instituto Nacional del Cáncer. (2019). Diccionario de cáncer: medicina personalizada de precisión. Retrieved April 29, 2019, from https://www.cancer.gov/espanol/publicaciones/diccionario/def/medicina-personalizada

Islam, N., & Rehman, A. (2013). *A Comparative Study of Major Service Providers for Cloud Computing*. Retrieved from https://www.researchgate.net/publication/258489864_A_Comparative_Study_of_Major_Service_Providers_for_Cloud_Computing

Jeffares, D. C., Jolly, C., Hoti, M., Speed, D., Shaw, L., Rallis, C., … Sedlazeck, F. J. (2017). Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nature Communications*, *8*(1), 14061. https://doi.org/10.1038/ncomms14061

Karczewski, K. J., Fernald, G. H., Martin, A. R., Snyder, M., Tatonetti, N. P., & Dudley, J. T. (2014). STORMSeq: An Open-Source, User-Friendly Pipeline for Processing Personal Genomics Data in the Cloud. *PLoS ONE*, *9*(1), e84860. https://doi.org/10.1371/journal.pone.0084860

Kosugi, S., Momozawa, Y., Liu, X., Terao, C., Kubo, M., & Kamatani, Y. (2019). Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biology*, *20*(1), 117. https://doi.org/10.1186/s13059-019-1720-5

Kulkarni, P., & Frommolt, P. (2017). Challenges in the Setup of Large-scale Next-Generation

Sequencing Analysis Workflows. *Computational and Structural Biotechnology Journal*, *15*, 471–477. https://doi.org/10.1016/j.csbj.2017.10.001

Kumbrink, J., & Kirsch, K. H. (2012). Regulation of p130(Cas)/BCAR1 expression in tamoxifen-sensitive and tamoxifen-resistant breast cancer cells by EGR1 and NAB2. *Neoplasia (New York, N.Y.)*, *14*(2), 108–120. https://doi.org/10.1593/neo.111760

Langmead, B., & Nellore, A. (2018). Cloud computing for genomic data analysis and collaboration. *Nature Reviews Genetics*, *19*(4), 208–219. https://doi.org/10.1038/nrg.2017.113

Li, A., Yang, X., Kandula, S., & Zhang, M. (2010). CloudCmp. *Proceedings of the 10th Annual Conference on Internet Measurement - IMC '10*, 1. https://doi.org/10.1145/1879141.1879143

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, *25*(14), 1754–1760. https://doi.org/10.1093/bioinformatics/btp324

M. Lesk, A. (2019). Bioinformatics | science | Britannica.com. Retrieved April 29, 2019, from https://www.britannica.com/science/bioinformatics

Martincorena, I., & Campbell, P. J. (2015). Somatic mutation in cancer and normal cells. *Science (New York, N.Y.)*, *349*(6255), 1483–1489. https://doi.org/10.1126/science.aab4082

Medina-Martínez, J. S., Arango-Ossa, J. E., Gundem, G., Levine, M. F., Patel, M., Farnoud, N. R., … Papaemmanuil, E. (2019). Abstract 5105: A plug-and-play infrastructure for scalable bioinformatics operations. In *Bioinformatics, Convergence Science, and Systems Biology*. https://doi.org/10.1158/1538-7445.AM2019-5105

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.SP.800-145

Memorial Sloan Kettering Cancer Center. (2019a). MSK-IMPACT: A Targeted Test for Mutations in Both Rare and Common Cancers | Memorial Sloan Kettering Cancer Center. Retrieved October 13, 2019, from https://www.mskcc.org/msk-impact

Memorial Sloan Kettering Cancer Center. (2019b). Targeted sequencing of adenomyoepithelioma. Retrieved October 15, 2019, from https://www.ebi.ac.uk/ena/data/view/PRJNA299807

Microsoft Azure. (2019). Pricing Calculator. Retrieved October 15, 2019, from https://azure.microsoft.com/en-us/pricing/calculator/

Nagahashi, M., Shimada, Y., Ichikawa, H., Kameyama, H., Takabe, K., Okuda, S., & Wakai, T. (2019). Next generation sequencing-based gene panel tests for the management of solid tumors. *Cancer Science*, *110*(1), 6. https://doi.org/10.1111/CAS.13837

National Center for Biotechnology Information. (2019). GRCh38.p13 - Genome - Assembly. Retrieved from https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39

National Center for Biotechnology Information (NCBI). (2017). Overview of Structural Variation. Retrieved October 11, 2019, from https://www.ncbi.nlm.nih.gov/dbvar/content/overview/

National Human Genome Research Institute (NHGRI). (2018). DNA Sequencing Costs: Data | NHGRI. Retrieved April 29, 2019, from https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data

National Institute for Computational Sciences, U. of T. (n.d.). What is HPC? Retrieved April 29, 2019, from https://www.nics.tennessee.edu/computing-resources/what-is-hpc

NetApp. (2019). What Is High-Performance Computing (HPC)? | How It Works. Retrieved April 29, 2019, from https://www.netapp.com/us/info/what-is-high-performance-computing.aspx

Nexica - Econocom Group. (2015). Modelos de despliegue cloud: Cloud privado, cloud público y cloud híbrido | Nexica. Retrieved April 29, 2019, from https://www.nexica.com/es/blog/modelos-de-despliegue-cloud-cloud-privado-cloud-público-y-cloud-híbrido

NIH- National Cancer Institute. (n.d.). Definition of DNA - NCI Dictionary of Genetics Terms. Retrieved April 29, 2019, from https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/dna

NIH- National Cancer Institute. (2017). The Genetics of Cancer. Retrieved April 29, 2019, from https://www.cancer.gov/about-cancer/causes-prevention/genetics

NIH- National Human Genome Research Institute. (2013). Bioinformatics: Introduction. Retrieved April 29, 2019, from https://www.genome.gov/25020000/online-education-kit-bioinformatics-introduction

NIH- National Human Genome Research Institute. (2015). A Brief Guide to Genomics. Retrieved April 29, 2019, from https://www.genome.gov/about-genomics/fact-sheets/A-Brief-Guide-to-Genomics

NIH- National Library of Medicine. (2019). What is a gene mutation and how do mutations occur? - Genetics Home Reference. Retrieved April 29, 2019, from https://ghr.nlm.nih.gov/primer/mutationsanddisorders/genemutation

NIH- U.S National Library of Medicine. (2019a). What is a gene? - Genetics Home Reference. Retrieved April 29, 2019, from https://ghr.nlm.nih.gov/primer/basics/gene

NIH- U.S National Library of Medicine. (2019b). What is DNA? - Genetics Home Reference. Retrieved April 29, 2019, from https://ghr.nlm.nih.gov/primer/basics/dna

O'Connor, B. (2014). Next-Generation Sequencing Analysis on the Grid and in the Cloud.

Retrieved October 11, 2019, from https://seqware.github.io/

Papenfuss Lab. (2019). GRIDSS: the Genomic Rearrangement IDentification Software Suite. Retrieved from https://github.com/PapenfussLab/gridss

Perrin, H. (2017). Best bioinformatics software for circos plot generation - omicX. Retrieved October 22, 2019, from https://omictools.com/blog/your-top-3-circos-plot-generation-tools

Rausch, T., Zichner, T., Schlattl, A., Stütz, A. M., Benes, V., & Korbel, J. O. (2012). DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics (Oxford, England)*, *28*(18), i333–i339. https://doi.org/10.1093/bioinformatics/bts378

Reid, J. G., Carroll, A., Veeraraghavan, N., Dahdouli, M., Sundquist, A., English, A., … Boerwinkle, E. (2014). Launching genomics into the cloud: Deployment of Mercury, a next generation sequence analysis pipeline. *BMC Bioinformatics*, *15*(1), 1–11. https://doi.org/10.1186/1471-2105-15-30

Roy, S., Coldren, C., Karunamurthy, A., Kip, N. S., Klee, E. W., Lincoln, S. E., … Carter, A. B. (2018). Standards and Guidelines for Validating Next-Generation Sequencing Bioinformatics Pipelines: A Joint Recommendation of the Association for Molecular Pathology and the College of American Pathologists. *The Journal of Molecular Diagnostics*, *20*(1), 4–27. https://doi.org/10.1016/J.JMOLDX.2017.11.003

Sadedin, S. P., Pope, B., & Oshlack, A. (2012). Bpipe: a tool for running and managing bioinformatics pipelines. *Bioinformatics*, *28*(11), 1525–1526. https://doi.org/10.1093/bioinformatics/bts167

Schramedei, K., Mörbt, N., Pfeifer, G., Läuter, J., Rosolowski, M., Tomm, J. M., … Brocke-Heidrich, K. (2011). MicroRNA-21 targets tumor suppressor genes ANP32A and SMARCA4. *Oncogene*, *30*(26), 2975–2985. https://doi.org/10.1038/onc.2011.15

Sedlazeck, F. (2019). SURVIVOR. Retrieved October 13, 2019, from https://github.com/fritzsedlazeck/SURVIVOR/wiki

Seven Bridges Genomics. (2019). The Seven Bridges Platform. Retrieved October 11, 2019, from https://www.sevenbridges.com/platform/

Shang, J., Zhu, F., Vongsangnak, W., Tang, Y., Zhang, W., & Shen, B. (2014). Evaluation and comparison of multiple aligners for next-generation sequencing data analysis. *BioMed Research International*, *2014*, 309650. https://doi.org/10.1155/2014/309650

Singh, P. (2018). A Guide To Agile Scrum Methodology in Mobile App Development. Retrieved October 15, 2019, from https://appinventiv.com/blog/agile-scrum-methodology-in-mobile-app-development/

Souilmi, Y., Lancaster, A. K., Jung, J.-Y., Rizzo, E., Hawkins, J. B., Powles, R., … Wall, D. P. (2015). Scalable and cost-effective NGS genotyping in the cloud. *BMC Medical*

Genomics, *8*. https://doi.org/10.1186/S12920-015-0134-9

SSH Communications Security, I. (2018). Cloud computing models (IaaS, PaaS & SaaS) | SSH.COM. Retrieved April 29, 2019, from https://www.ssh.com/cloud/computing/models#sec-Based-on-Service-Model-Architecture-and-Flexibility

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., … Robinson, G. E. (2015). Big Data: Astronomical or Genomical? *PLoS Biology*, *13*(7), e1002195. https://doi.org/10.1371/journal.pbio.1002195

Straiton, J., Free, T., Sawyer, A., & Martin, J. (2019). From Sanger sequencing to genome databases and beyond. *BioTechniques*, *66*(2), 60–63. https://doi.org/10.2144/btn-2019-0011

tecnoiver ltda. (2015). Servicios en la nube y modelos de servicio. Retrieved April 29, 2019, from https://www.tecnoinver.cl/servicios-en-la-nube-y-modelos-de-servicio/

Tefferi, A. (2006). Genomics Basics: DNA Structure, Gene Expression, Cloning, Genetic Mapping, and Molecular Tests. *Seminars in Cardiothoracic and Vascular Anesthesia*, *10*(4), 282–290. https://doi.org/10.1177/1089253206294343

The Apache Software Foundation. (2018). What is a Workflow Management System? Retrieved October 11, 2019, from https://taverna.incubator.apache.org/introduction/what-is-a-workflow-management-system

Tuma, R. S. (2012, January 18). ALK gene amplified in most inflammatory breast cancers. *Journal of the National Cancer Institute*, Vol. 104, pp. 87–88. https://doi.org/10.1093/jnci/djr553

University of California. (2018). UCSC Genome Browser Downloads. Retrieved October 13, 2019, from https://hgdownload.soe.ucsc.edu/downloads.html

Verma, M. (2012). Personalized medicine and cancer. *Journal of Personalized Medicine*, *2*(1), 1–14. https://doi.org/10.3390/jpm2010001

Visma AS. (2019). Cloud basics – Deployment models | Corporate Blog. Retrieved April 29, 2019, from https://www.visma.com/blog/cloud-basics-deployment-models/

Wagle, P., Nikolić, M., & Frommolt, P. (2015). QuickNGS elevates Next-Generation Sequencing data analysis to a new level of automation. *BMC Genomics*, *16*(1), 487. https://doi.org/10.1186/s12864-015-1695-x

Wheeler, D. A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A., … Rothberg, J. M. (2008). The complete genome of an individual by massively parallel DNA sequencing. *Nature*, *452*(7189), 872–876. https://doi.org/10.1038/nature06884

Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., … Goble, C.

(2013). 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. https://doi.org/10.1093/nar/gkt328

Xtelligent Healthcare Media, L. (2018). What Are Precision Medicine and Personalized Medicine? Retrieved October 21, 2019, from https://healthitanalytics.com/features/what-are-precision-medicine-and-personalized-medicine

Yi, K., & Ju, Y. S. (2018). Patterns and mechanisms of structural variations in human cancer. *Experimental & Molecular Medicine*, *50*(8), 98. https://doi.org/10.1038/s12276-018-0112-3

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18. https://doi.org/10.1007/s13174-010-0007-6

Zhang, Z., Lin, D., Xin, G., Yan, G., & Jingfa Xiao. (2012). Bioinformatics clouds for big data manipulation. *Biology Direct*, *7*, 43; discussion 43. Retrieved from http://www.biologydirect.com/content/7/1/43

Zhao, S., Prenger, K., Smith, L., Messina, T., Fan, H., Jaeger, E., & Stephens, S. (2013). Rainbow: A tool for large-scale whole-genome sequencing data analysis using cloud computing. *BMC Genomics*, *14*(1), 1. https://doi.org/10.1186/1471-2164-14-425