Pontificia Universidad JAVERIANA Cali

Facultad de Ingenieria y Ciencias

Acta de Correcciones al Proyecto de Grado
Biología

Fecha: 28 de enero de 2021

Autores: Giann Karlo Aguirre Samboní

Nombre del Proyecto de Grado: Construcción de un Mapa Genético para una Variedad Comercial de Caña de Azúcar (*Saccharum* spp.)

Director: John Jaime Riascos Arcos

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

_____

Firma del Director del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Biólogo.

Dr. HERNAN CAMILO ROCHA NIÑO
Decano Facultad de Ingeniería

DR. MATEO LOPEZ VICTORIA
Director Carrera Biología

John Jaime Riascos Arcos
Director Trabajo de Grado

Nombre Jurado 1
Fabián Tobar Tosse

Nombre Jurado 1
Diego Riaño Pachón

Santiago de Cali, 7 de diciembre de 2020

cenicaña

Centro de Investigación
de la Caña de Azúcar
de Colombia

Doctor
Mateo López Victoria
Director de la carrera de Biología
Pontificia Universidad Javeriana
Cali, Colombia

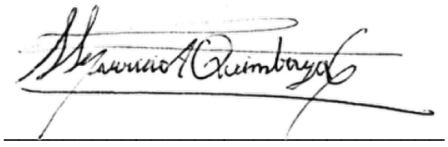
Cordial saludo,

Por medio de la presente certifico que el trabajo de grado titulado "Construcción de un mapa genético para una variedad de caña de azúcar (*Saccharum* spp.)" realizado por el estudiante Giann Karlo Aguirre Samboní con el código 0072857, estudiante de la carrera de Biología en la Facultad de Ingeniería y Ciencias de la Pontificia Universidad Javeriana Cali, se encuentra terminado y puede ser presentado para sustentación. Además, confirmo que he asumido la dirección en reemplazo de Mauricio Alberto Quimbaya Gómez, quien asumió la codirección de este trabajo de grado.

Atentamente,

_____

John Jaime Riascos Arcos
Director tesis
Centro de Investigación de la Caña de Azúcar de Colombia, CENICAÑA
Cali, Colombia
Tel: +57 (2) 5246611 Ext: 5127
jjriascos@cenicana.org



_____

Mauricio Alberto Quimbaya Gómez
Codirector tesis
Pontificia Universidad Javeriana
Cali, Colombia
Tel: +57 (2) 3218200 Ext: 8636
maquimbaya@javerianacali.edu.co

Santiago de Cali, Diciembre 7 2020

Doctor
Mateo López Victoria
Director de la carrera de Biología
Pontificia Universidad Javeriana
Cali, Colombia

Cordial saludo,

Me permito presentar el proyecto de grado "Construcción de un mapa genético para una variedad de caña de azúcar (*Saccharum* spp.)" a la Facultad de Ingeniería y Ciencias para asignación de evaluadores y definición de fecha de sustentación, con el fin de cumplir con los requisitos parcial exigidos por la universidad para optar por el título de Biólogo.

Atentamente,

Giann Karlo Aguirre Samboní

_____

Giann Karlo Aguirre Samboní

Cód. 0072857

Centro de Investigación de la Caña de Azúcar de Colombia

Construcción de un mapa genético para una variedad comercial de caña de azúcar (*Saccharum* spp.)

Giann Karlo Aguirre Samboní

Pontificia Universidad Javeriana
Facultad de Ingeniería y Ciencias
Biología
Santiago de Cali
2021

Centro de Investigación de la Caña de Azúcar de Colombia

Construcción de un mapa genético para una variedad comercial de caña de azúcar (*Saccharum* spp.)

Giann Karlo Aguirre Samboní

Trabajo de grado presentado para optar al título de Biólogo

Director:
John Jaime
Riascos Arcos

Co-director:
Jorge Alexander
Duitama Castellanos

Co-director:
Mauricio Alberto
Quimbaya Gómez

Pontificia Universidad Javeriana
Facultad de Ingeniería y Ciencias
Biología
Santiago de Cali
2021

A mi mamá y mis hermanos por
su constante coherencia,
estricta disciplina y,
sobre todo, su amor incondicional.

Abstract

A genetic map is compounded by molecular markers that are separated by relative genetic distances due to crosses between individuals. Genetic maps are used in the assembly of complex genomes such as sugarcane hybrids (*Saccharum* spp.) because they have highly heterozygous, autopolyploid and aneuploid genomes. The sugarcane cultivar CC 01-1940 has been used to generate long sequences called scaffolds and a potential genetic map will serve as a guide of the scaffolds in order to depict the linkage groups or chromosomes. This research shows the construction of three genetic maps using different strategies whose data comes from high-throughput next-generation sequencing (NGS) technologies of a biparental F1 population. The progeny is a result of the crossing between sugarcane cultivars CC 01-746 and CC 01-1940. The 93 individuals plus both parents were sequenced using Whole Genome Sequencing (WGS). This set of reads per individual were aligned to the scaffolds with outcomes higher than 85%. The alignments were used for variants detection and selection of high quality SNPs that enable linkage map construction. A total of 3,739 loci, 4,096 loci and 7,816 loci were obtained respectively for each SNP calling strategy. Strategy A yielded a map size of 9,679.50 cM with an average density of 6.05 cM/loci. Strategy B yielded a map of 1,191.30 cM with an average density of 1.96 cM/loci and strategy C yielded a map of 1,198.74 cM with an average density of 2.19 cM/loci. This is the first time that CENICAÑA builds genetic maps from NGS data, which releases an important pipeline to advance in the genome assembly and unveils a tool for future work on genotype-phenotype association. The three genetic maps presented here are able to assemble linkage groups from the cultivar scaffolds. This methodology contributes significantly to CENICAÑA's platform of molecular marker assisted breeding.

Resumen

Un mapa genético está compuesto por marcadores moleculares que están separados por distancias genéticas relativas debido a los cruces entre individuos. Los mapas genéticos son usados en el ensamblaje de genomas complejos como los híbridos de la caña de azúcar (*Saccharum* spp.) porque son genomas altamente heterocigotos, autopoliploides y aneuploides. Para la variedad de caña CC 01-1940 se han generado unas secuencias largas llamadas *scaffolds* y el mapa genético servirá como anclaje de los *scaffolds* para llegar al nivel de grupos de ligamiento o cromosomas. En este trabajo se muestra la construcción de tres mapas genéticos usando diferentes estrategias a partir de secuenciación de alto rendimiento de una población F1 biparental, resultado del cruce entre las variedades de caña de azúcar CC 01-746 y CC 01-1940. De la progenie fueron secuenciados el genoma completo (WGS) de 93 individuos más ambos padres. Este conjunto de lecturas por individuo fueron alineadas a los *scaffolds* con resultados superiores al 85 %. Los alineamientos fueron usados para la detección de variantes y selección de marcadores moleculares SNPs de calidad suficiente para construir los mapas. Un total de 3,739 loci, 4,096 loci y 7,816 loci fueron obtenidos respectivamente para cada estrategia de llamado de SNPs. La estrategia A arrojó un tamaño del mapa de 9,679.50 cM con una densidad promedio de 6.05 cM/loci; la estrategia B resultó en un mapa de 1,191.30 cM con una densidad promedio de 1.96 cM/loci; y la estrategia C generó un mapa de 1,198.74 cM con una densidad promedio de 2.19 cM/loci. Esta es la primera vez que CENICAÑA construye mapas genéticos a partir de datos de secuenciación masiva, lo que significa un *pipeline* importante para avanzar en el ensamblaje del genoma que esta en construcción y para trabajos futuros de asociación genotipo-fenotipo. Los tres mapas genéticos construidos tienen el potencial de ensamblar grupos de ligamiento a partir de los *scaffolds* de la variedad. Esta metodología contribuye significativamente a la plataforma de mejoramiento asistida por marcadores moleculares de CENICAÑA.

Índice general

Índice de figuras

Índice de tablas

Introducción

La caña de azúcar (*Saccharum* spp.) es una planta de interés comercial y es un cultivo energético de alta biomasa cuyo residuo lignocelulósico, después de la extracción del azúcar, es usado para la producción de biocombustible u otros bioproductos (Awe *et al.*, 2020). Este cultivo es responsable del 85 %-87 % de la producción mundial de azúcar mientras la restante es producida a partir de remolacha (*Beta* spp.) (OECD y FAO, 2018). La mayoría de cultivos de caña de azúcar están en países tropicales y subtropicales (Kapoor *et al.*, 2016).

Los grandes consumidores de azúcar en el mundo son también los más grandes productores: Brasil, India, China, Tailandia, Pakistán, México y Colombia (Miranda y Fonseca, 2020). En Colombia, el cultivo se encuentra ubicado en el valle geográfico del río Cauca que abarca desde el norte del departamento del Cauca hasta el sur del departamento de Risaralda. En esta región hay 238,134 hectáreas sembradas de variedades de caña (Asocaña, 2019) en tres tipos de ambientes: semi seco, húmedo y piedemonte. El 90 % de las variedades sembradas en estos ambientes provienen del programa de mejoramiento genético del Centro de Investigación de la Caña de Azúcar de Colombia, CENICAÑA (Asocaña, 2019).

Desde el año 2000, CENICAÑA y su programa de mejoramiento realizan el desarrollo de variedades para la agroindustria de la caña de Colombia. A partir de la caracterización y evaluación del Banco de germoplasma, en cada uno de los tres ambientes se selecciona un grupo élite de variedades que sirve de base para la programación y realización de los cruzamientos. Este proceso busca el complemento genético de características fenotípicas de interés, entre las variedades que intervienen en un cruzamiento dado (Viveros, 2018). La metodología del programa ha producido resultados significativos para mejorar la productividad y el rendimiento de la agroindustria azucarera de Colombia, a pesar de que los tiempos de cosecha (10-12 meses) y de generación de una nueva variedad (10-15 años) sean extensos (Donzelli *et al.*, 2018).

Desde 1980 a la actualidad, el Centro ha generado 65 variedades (ver listado en la tabla A.1) de caña comercial para satisfacer las necesidades de siembra en los diferentes ambientes (Arango *et al.*, 2018; Borrero *et al.*, 2019; Victoria *et al.*, 2013). Recientemente, el Centro ha venido trabajando en una plataforma de mejoramiento asistida por marcadores moleculares, conocida como proyecto SAM (Selección Asistida por Marcadores Moleculares). Esta plataforma es útil para acelerar los tiempos que se toma la acumulación de ganancias genéticas en las variedades, las cuales quedan representadas en progenies sobresalientes. Una parte del proyecto SAM incluye el ensamblaje del genoma de un híbrido comercial de CENICAÑA, el cual sería una pieza fundamental para poder explotar la diversidad genética de la especie mediante la identificación de marcadores moleculares de alta calidad. Además, los marcadores moleculares le servirían al Centro para diferentes estudios incluidos el análisis de asociación fenotipo-genotipo (*Genome Wide Association Studies*, GWAS). Sin embargo, a falta de un genoma de caña, el hallazgo de marcadores moleculares se ha hecho antes con el genoma de sorgo (*Sorghum bicolor*).

La caña de azúcar es un autopoliploide que ha tenido dos eventos de duplicaciones en todo el genoma que se produjeron hace 1-2 millones de años (Jannoo *et al.*, 2007). No obstante, los

híbridos comerciales de caña de azúcar (*Saccharum* spp.) están compuestos por dos sub-genomas provenientes de dos especies, la hembra, con alto contenido de azúcar de tallo ancho, *Saccharum officinarum* (2n= $8\times = 80, x = 10$) y el macho, de tallo delgado con poco azúcar, *S. spontaneum* (2n= $5\times = 40$ a $16\times = 128, x = 8$) (D'Hont, Grivet *et al.*, 1996; D'Hont, Ison *et al.*, 1998; D'Hont y Glazsmann, 2001; J. Zhang, Nagai *et al.*, 2012). El genoma de los híbridos comerciales es complejo por tener una recombinación desigual de características genéticas heredadas de ambos parentales, es decir, 80 % de los cromosomas vienen de *S. officinarum*, 10-15 % de los cromosomas vienen de *S. spontaneum* y 5-10 % es material recombinante de ambos (Cuadrado *et al.*, 2004; D'Hont, 2005; D'Hont, Grivet *et al.*, 1996; G. Piperidis *et al.*, 2010). Por lo tanto, los híbridos de caña de azúcar son altamente heterocigotos, tienen un alto nivel de ploidía, alto contenido de ADN repetitivo y presentan aneuploidía en sus grupos homoeólogos de cromosomas (Ming *et al.*, 1998; Mohan, 2016).

Esta complejidad en el genoma de los híbridos hace que este cultivo, a pesar de ser económicamente importante, haya quedado atrás en estudios de genómica con respecto a otros tipos de cultivos como el arroz, la cebada, el trigo, el maíz, entre otros (Grivet y Arruda, 2002; Thirugnanasambandam *et al.*, 2018). El ensamblaje del genoma de un híbrido comercial sería un avance significativo para proyectos como SAM en CENICAÑA, aún así cuando el genoma de *S. spontaneum* ya fue ensamblado (J. Zhang, X. Zhang *et al.*, 2018). El tamaño total del genoma nuclear de la caña de azúcar moderna es alrededor de 10.0 Gb, pero su genoma monoploide tiene un tamaño ca. 1.0 Gb (D'Hont y Glazsmann, 2001; Le Cunff *et al.*, 2008). El objetivo inicial de CENICAÑA es ensamblar el genoma monoploide para la variedad CC 01-1940.

En 1999, CENICAÑA efectuó el cruzamiento entre las variedades CCSP 89-1997 y CC 91-1583 y como resultado de la progenie de este cruce, uno de los individuos fue nombrado como CENICAÑA Colombia (CC) 01-1940. A través de los estados de selección en ambiente húmedo, esta variedad superó en producción de toneladas de sacarosa por hectárea (TSH), entre 1.8 % a 31.0 %, a la CC 85-92, la que en su momento tenía una alta productividad y estaba sembrada en más del 60 % de la región (Viveros, 2018). A partir de 2010, la variedad CC 01-1940 empezó su comercialización y siembra en ambiente húmedo en los ingenios de la región, los resultados mostraron que esta variedad tenía mejores indicadores de toneladas de caña por hectárea (TCH), sacarosa (TSH) y porcentaje de rendimiento en azúcar que la CC 85-92. Hoy en día la CC 01-1940 se encuentra sembrada en aproximadamente 80.000 hectáreas y, en los últimos cinco años, sobrepasa en 2 toneladas de azúcar por hectárea los resultados de la variedad CC 85-92 (Viveros, 2018), lo que hace a CC 01-1940 candidata para estudiar y armar su genoma.

El ensamblaje de un genoma empieza con la secuenciación del ADN de una especie o variedad de interés, en este caso, CC 01-1940. Dada la complejidad de este híbrido de caña, fue necesario hacer uso de tres tipos de tecnologías de secuenciación: (1) lecturas largas de PacBio, poca fragmentación pero baja calidad (5-15 % de error) (Rhoads y Au, 2015), (2) lecturas cortas pareadas de Illumina, alta calidad (0.1 % de error) pero alta fragmentación por regiones repetitivas (Bansal y Boucher, 2019; Meyer y Kircher, 2010) y (3) lecturas cortas de Illumina usando la metodología Hi-C, que permite la identificación y purificación de interacciones de cromatina en un genoma completo seguido de una secuenciación masiva en paralelo (Lieberman-Aiden *et al.*, 2009). Esta última metodología Hi-C permitió llevar la continuidad del ensamblaje de contigs (secuencias cortas ensambladas) a *scaffolds* (secuencias largas ensambladas), sin embargo, las tecnologías de secuenciación para un genoma complejo no alcanzan a construir cromosomas y a menudo se requiere de estrategias adicionales para lograrlo, por ejemplo, los mapas genéticos.

El mapeo génico es la asignación secuencial de loci a una posición relativa en un cromosoma. Los mapas genéticos son específicos de cada especie, y están compuestos de marcadores genómicos o genes y la distancia genética entre cada marcador. Estas distancias se calculan en función de la frecuencia de los cruces cromosómicos que se producen durante la primera división de la meiosis,

profase I, y no de su ubicación física en el cromosoma (Saraswathy y Ramalingam, 2011). Para la construcción de un mapa genético es necesaria la variabilidad genética entre individuos de un grupo o una población para poder establecer los marcadores moleculares, dado que estos se vuelven estimadores de la presencia o ausencia de una característica de interés (Little, 2005).

En CENICAÑA se cruzaron las variedades CC 01-746 (la madre) y CC 01-1940 (el padre) las cuales tienen características contrastantes: la primera tiene menos TCH pero más producción de sacarosa, floración y resistencia a roya naranja (causada por el hongo *Puccinia kuehnii*), mientras que la segunda presenta indicadores opuestos en estas variables. De la progenie de este cruzamiento fueron seleccionados al azar 93 individuos y ambos padres, con el propósito de extraer el ADN y someterlos a un proceso de secuenciación masiva del genoma completo de cada uno. El resultado de las secuencias son el insumo para la detección de marcadores moleculares SNPs (*Single Nucleotide Polymorphisms*) y estos últimos son los componentes principales con los que se construye el mapa genético.

Los mapas genéticos ofrecen unidades de distancia basadas en la frecuencia de recombinación, centiMorgan (cM) o unidad de mapa (mu). Estas están definidas como la distancia entre dos marcadores moleculares para los que la recombinación ocurre con una frecuencia de 1 % (Cuschieri, 2018). Por tanto, 10 mu entre dos marcadores indican 10 % de frecuencia de recombinación entre ellos (10 % del tiempo, no serán heredados juntos, mientras que un 90 % lo harán) (Paz y Shoemaker, 2005). Al usar la tasa de recombinación se podría obtener una estimación de la separación física en pares de bases (Hultén y Tease, 2006).

Actualmente, existen diferentes herramientas bioinformáticas para construir un mapa genético entre ellas están polymapR (Bourke, van-Guesst *et al.*, 2018), PERGOLA (Grandke *et al.*, 2017), OneMap (Margarido *et al.*, 2007), netgwas (Behrouzi *et al.*, 2017) y JoinMap (Stam, 1993). Estas herramientas se basan en la segregación de los marcadores moleculares en la población para hacer un ordenamiento y establecer los grupos de ligamiento dentro del mapa. Un ordenamiento correcto en el mapa genético ayudará a hacer genética comparativa con el genoma de *S. spontaneum* (J. Zhang, X. Zhang *et al.*, 2018), ensamblar los *scaffolds* de CC 01-1940 y permitiría hacer estudios clásicos de QTLs (*Quantitative Trait Loci*), considerados como regiones genómicas que albergan los genes que rigen la expresión de un rasgo cuantitativo (Boopathi, 2020). Los QTLs son de interés porque se podrían identificar características favorables del cruzamiento como TCH, TSH o resistencia a enfermedades, por ejemplo, la roya naranja. En este trabajo se presenta la construcción de tres mapas genéticos para la variedad CC 01-1940 de CENICAÑA usando tres aproximaciones diferentes.

1.1. Objetivos

1.1.1. General

Construir un mapa genético para caña de azúcar (*Saccharum* spp.) que apoye el proceso de ensamblaje del genoma de la variedad comercial CC 01-1940.

1.1.2. Específicos

1. Caracterizar variantes genómicas entre los parentales CC 01-746 y CC 01-1940 a partir de datos de secuenciación del genoma completo (WGS) de una población F1 de caña de azúcar.

2. Desarrollar un *pipeline* para crear un mapa genético de la variedad comercial de caña de azúcar (*Saccharum* spp.) CC 01-1940.

Métodos

2.1. Material vegetal

Se usaron 95 genotipos de caña de azúcar para este estudio, 93 son hermanos completos provenientes de una población bi-parental cuyos parentales son los 2 genotipos restantes. La variedad CC 01-746 representó la madre al presentar menor polen viable y este híbrido, además, se caracteriza por tener menor TCH, mayor producción de sacarosa, mayor floración y mayor resistencia a roya naranja. La variedad CC 01-1940 fue el padre del cruzamiento con características contrastantes, es decir, mayor polen, mayor TCH, menor producción de sacarosa, menor floración y menor resistencia a roya naranja. El cruzamiento entre estas variedades generó 315 individuos, de los cuales fueron escogidas al azar 93 muestras. Este paso fue ejecutado previo al inicio de este trabajo.

2.2. Extracción de ADN

Se colectaron en nitrógeno líquido 100 mg de tejido foliar proveniente de las hojas jóvenes de cada genotipo. Cada una de estas muestras fue pulverizada y el ADN se extrajo siguiendo las indicaciones en Dellaporta *et al.* (1983), las cuales se adaptaron para facilitar la extracción en platos de 95 pozos. Cada muestra de ADN fue tratada con RNasa libre de DNasa (ThermoFischer Scientific, Waltham, MA, USA) y diluida en ddH_2O. La calidad del ADN se comprobó en un gel de agarosa al 0,8 % y se cuantificó con un espectrofotómetro NanoDrop (Thermofisher, EE.UU.). Aproximadamente 5 µg de cada muestra de ADN fueron enviados a la compañía NOVOGENE (Beijing, China) para la construcción de bibliotecas y la secuenciación de genoma completo (*whole genome sequencing*, WGS). La extracción se realizó previamente al inicio de este trabajo.

2.3. Secuenciación de las muestras

La empresa NOVOGENE (Beijing, China) realizó la secuenciación de segunda generación WGS, al inicio de este estudio, de los genotipos y fueron secuenciadas sus regiones codificantes y no codificantes. Las librerías de secuenciación se construyeron utilizando el equipo Illumina TruSeq DNA PCR-Free Kit para fragmentos de 500 pares de bases (bp), y se visualizaron en los chips de ADN de alta sensibilidad Agilent. Los sistemas HiSeq 2000 o HiSeq 2500 de Illumina fueron usados para secuenciar los *clusters* de los fragmentos (500 bp) para generar lecturas pareadas (*paired-end*) de 150 bp. Múltiples muestras fueron puestas en cada banda con el fin de alcanzar profundidades de secuenciación de al menos 50x por individuo del genoma completo y generar datos de secuenciación de alta calidad que aumenten la probabilidad de alineamiento.

2.4. Genoma de referencia

Se usó el ensamblaje de un híbrido colombiano CC 01-1940, variedad generada dentro del programa de mejoramiento genético de CENICAÑA. En la construcción de este genoma se usaron tres tipos de secuenciación: lecturas largas de PacBio, lecturas cortas pareadas de Illumina y lecturas cortas de Illumina usando la metodología de secuenciación Hi-C (Trujillo, 2020). A partir del conjunto de estas lecturas se generó el ensamblaje en términos de *scaffolds*, que representa el genoma monoploide de la variedad CC 01-1940. El genoma monoploide de esta variedad tiene un tamaño aproximado de 1,019 Mbp ($2n = 1\times = 10, x = 10$). Este ensamblaje se caracteriza por tener: un total de 17,098 *scaffolds*, la secuencia más larga ensamblada tiene un tamaño de 29.81 Mbp y la más pequeña es de 1 kbp, su N50 es de 1.23 Mbp y 186 secuencias tienen al menos este tamaño de N50 (Trujillo, 2020) (tabla 2.1).

Tabla 2.1: Medidas descriptivas del genoma de referencia, ensamblaje de CC 01-1940 a nivel de *scaffolds*.

Métrica	**Versión 3 (PacBio + Hi-C)**
# Scaffolds	17,098
Tamaño	1,253 Mbp
N50	1.23 Mbp
n:N50	186
Min	1,000 bp
Max	29.81 Mbp

2.5. Evaluación y alineación de las muestras

Los datos de secuenciación fueron evaluados con el software FastQC v11.8 (Andrews *et al.*, 2018). Por cada individuo se analizaron dos conjuntos con un gran número de secuencias, el primero con orientación *forward* y el otro con orientación *reverse*. Esta evaluación determina cuán confiable son los asignamientos para cada llamado en la detección de una base dentro del proceso de secuenciación (ver algoritmo A.1).

Posterior a la evaluación de calidad se calculó el tamaño del inserto que incluye la longitud de ambas lecturas (*forward-reverse*) más la distancia que hay entre estas (*inner distance*) (ver algoritmo A.2). El rango del tamaño del inserto se calculó de 0 a 600 bp para realizar el proceso de alineamiento al genoma de referencia. Varios *softwares* fueron usados para cumplir esta actividad: (1) Bowtie2 v2.3.5 (Langmead y Salzberg, 2012), usado como una herramienta para el alineamiento de las lecturas secuenciadas a una referencia de secuencias largas (*scaffolds*), (2) Picard v2.20 (*Picard Toolkit* 2019), usado como una herramienta para ordenar los alineamientos por coordenadas y (3) Samtools v1.9 (Li *et al.*, 2009), instrucción usada para guardar el archivo con el formato y datos necesarios (`.bam`) (ver algoritmo A.3).

2.6. Llamado de variantes y filtros

Con los archivos alineados y ordenados se ejecutó el descubrimiento o llamado de variantes. El software usado fue *Next Generation Sequencing Experience Platform* (NGSEP) v3.3.3 (Duitama *et al.*, 2014; Tello *et al.*, 2019). El gran conjunto de datos generados (>2.5 TeraBytes) se procesó con

la rutina tradicional (`FindVariants`→`MergeVariants`→`FindVariants`→`MergeVCF`) y parámetros para procesar datos WGS propuesta en la documentación de NGSEP v3.3.3 (Perea *et al.*, 2016):

1. `FindVariants:` esta instrucción hace un descubrimiento de variantes para crear un catálogo de las mismas. Se estableció una ploidía promedio de 10 para la población dada la descripción de poliploidía y aneuploidía para caña de azúcar. Se estableció una calidad mínima de genotipado (`minQuality`) para aceptar un llamado de Single Nucleotide Variant (SNV), este parámetro está definido como 1 menos la probabilidad posterior del genotipo (valor Phred) y fue de 40. Se estableció un valor máximo permitido para el parámetro de calidad (`maxBaseQS`) en una base de 30, lo cual busca reducir el efecto de los errores de secuenciación con valores altos por puntaje de calidad. Este paso se ejecuta por cada archivo `.sorted.bam` de cada individuo. A partir de este paso se trabaja con archivos Variant Call Format (`.vcf`) (ver algoritmo A.4).

2. `MergeVariants:` esta función hace una unión de variantes que se descubre en cada individuo del paso anterior y se ejecuta una conciliación de alelos para las variantes intersectadas. Este paso genera un archivo en el que están todas las posiciones en una lista para poder genotipar cada muestra en las mismas posiciones y compararlas entre ellas. Los archivos que recibe de entrada son: el listado de nombres de las secuencias tal como aparecen en la referencia (`seq_names.txt`), para este caso, la identificación única de los *scaffolds*, por ejemplo, `scaffold_12997`, y archivos `.vcf` resultado del paso anterior (ver algoritmo A.5).

3. `FilterVCF:` esta orden hace un filtro al archivo del `MergeVariants` dado el gran conjunto de datos para hacer una reducción de variantes y acelerar el proceso. Estos filtros se ejecutaron para seleccionar marcadores moleculares SNPs bialélicos que (1) estén a 20 pares de bases de distancia, (2) en *scaffolds* de más de 100 kbp que no estuvieran en regiones de sintenia con el genoma de sorgo (análisis de sitenia previo a este trabajo), o (3) en regiones no repetitivas (ver algoritmo A.7).

4. `FindVariants:` este comando hace una genotipificación de los SNPs enlistados y descubiertos en los pasos anteriores con el fin de saber si cada individuo es heterocigoto, homocigoto al alelo de referencia u homocigoto al alelo alternativo para un loci. Se usaron los mismos parámetros para la calidad mínima de genotipado, el valor máximo de calidad en una base y el nivel de ploidía del `FindVariants` en el paso 1 (ver algoritmo A.8).

5. `MergeVCF:` la genotipificación anterior entrega un archivo `.vcf` por cada individuo y con esta instrucción se busca mezclar y consolidar todos los datos genotipados de la población en un único archivo. Para este paso se requiere también el archivo `seq_names.txt` (ver algoritmo A.9).

6. A partir de este paso, las estrategias, A, B y C se dividieron para poner a prueba sus hallazgos con diferentes filtros, una de ellas lo que se indicaba en Garsmeur *et al.* (2018). Estas estrategias son excluyentes entre sí para filtrar los marcadores y fueron ejecutadas así:

    A. `FilterVCF:` con esta instrucción se aplican filtros para buscar calidad de los marcadores y que sirvan para la construcción del mapa genético. Se seleccionaron los SNPs que estuvieran genotipados en al menos 60 individuos (`minI = 60`), es decir, que podían haber datos perdidos en 35 individuos. El *Minor Allele Frequency* (MAF) se fijó en al menos 0.10 (`minMAF=0.10`) para diferenciar que una variante se mantenga en al menos 10 % de la población. Se estableció un rango para la heterocigosidad observada (OH) de mayor o igual 0.10 (`minOH=0.10`) y menor o igual a 0.90 (`maxOH=0.90`) que garantiza

omitir las heterocigosidades extremas en un sistema diploide (ver algoritmo A.10, A.11 A.12).

I. `Algoritmo:` esta implementación se usó para filtrar los SNPs que no tienen genotipificación en al menos uno de los padres o que son SNPs homocigotos en ambos padres (ver algoritmo A.13).
   i. `Algoritmo:` esta ejecución buscó seleccionar los SNPs, de un análisis de diversidad de la población, que no tienen una desviación significativa del equilibrio Hardy-Weinberg (HWE). Es decir, el marcador es seleccionado cuando tiene un valor P mayor de 0.1 (ver algoritmo A.15). El producto de este archivo se usa en el paso II.
   ii. `FilterVCF:` se aplicaron filtros más exigentes en el MAF cuyo rango se estableció de 0.1 a 0.4 y el OH se asignó entre 0.2 y 0.8 (ver algoritmo A.14).

II. `FilterVCF:` a aquellos SNPs sin desviación significativa del HWE, aquellos generados del paso A.I.i., se les hizo la intersección con los SNPs que resultaron después del paso A.I.ii. Es decir, solo los SNPs que cumplieran ambas condiciones fueron seleccionados para generar el `.vcf` final que sería la entrada al *software* de construcción del mapa genético (ver algoritmo A.16).

B. Así como en la estrategia A, la instrucción `FilterVCF` también se ejecutó usando los mismos parámetros con los mismos valores. Es decir, se usaron `minI = 60`, `minMAF=0.10`, `minOH=0.10` y `maxOH=0.90` (ver algoritmo A.12). Algunos filtros posteriores sí fueron diferentes en sus valores pero reusando los algoritmos ya implementados en la Estrategia A:

I. Un algoritmo para filtrar los SNPs sin genotipificación en al menos uno de los padres u homocigotos en ambos padres (ver algoritmo A.13).
   i. Un algoritmo que selecciona aquellos SNPs que no tienen una desviación significativa del HWE. Esta vez es cuando el valor P es mayor de 0.01 (ver algoritmo A.15).
   ii. `FilterVCF:` a diferencia de la estrategia A, se aplicaron filtros más exigentes en el MAF cuyo rango se estableció de 0.2 a 0.4 y el OH se asignó entre 0.35 y 0.65 (ver algoritmo A.14).

II. `FilterVCF:` igual a la estrategia A, a aquellos SNPs sin desviación significativa del HWE, generados del paso B.I.i., se les hizo la intersección con los SNPs que resultaron después del paso B.I.ii. Es decir, solo los SNPs que cumplieran ambas condiciones fueron seleccionados para generar el `.vcf` final que sería la entrada al *software* de construcción del mapa genético (ver algoritmo A.16).

C. Esta estrategia es implementada y seguida por Garsmeur *et al.* (2018). La cual fue reproducida por un algoritmo que buscó filtrar todos aquellos SNPs en el padre 1 (CC 01-1940) que fueran heterocigotos (0/1) y todos los SNPs que en el padre 2 (CC 01-746) fueran homocigotos al alelo de referencia (0/0) o al alelo alternativo (1/1) (ver algoritmo A.17). Aquellos SNPs que no cumplían ambas condiciones, fueron descartados. Después se siguieron estos pasos:

I. `Algoritmo:` esta implementación buscó contar el número de individuos que, por marcador, eran heterocigotos (`ohet`) u homocigotos (`ohom`). El número de datos pérdidos (./.) fueron también contabilizados (`mis`) para ser descontados del total (`tot`) de los 95 individuos. Por ejemplo, si `ohet=63`, `ohom=22`, `tot=ohet+ohom=85` y `mis=10`. Estas variables de conteo por cada marcador fueron arrojadas, a modo

de tabla, en un archivo de texto plano: `ohet` (primera columna), `ohom` (segunda columna), `mis` (tercera columna) y `tot` (cuarta columna). El `.txt` final sería la entrada para hacer uso de una hoja de cálculo en la que se pueda hacer fácilmente una prueba chi cuadrado (ver algoritmo A.18).

II. `Hoja de cálculo:` posterior al archivo generado en el paso anterior, se utilizó un *software* para manipular hojas de cálculo, por ejemplo, *Microsoft Excel*. Las cuatro columnas generadas fueron pegadas a la hoja de cálculo y se le anexaron tres columnas más a la tabla: heterocigosidad esperada, la cual es igual para todos los marcadores (`ehet=42.5`), homocigosidad esperada, igual también para todos los marcadores (`ehom=42.5`) y, la última columna, el valor P de una prueba Chi cuadrado, por cada marcador, usando los valores observados con respectos a los esperados de la segregación (ver tabla A.3). En *Microsoft Excel*: `CHISQ.TEST(A2:B2,E2:F2)` donde `A2` es `ohet`, `B2` es `ohom`, `E2` es `ehet` y `F2` es `ehom`.

III. `Algoritmo:` dadas las siete columnas creadas, se implementó un algoritmo que seleccionará solo aquellos SNPs cuyo valor P fuera mayor o igual a 0.1, no tuvieran un sesgo significativo de los valores esperados. Es decir, no se desviarán significativamente de una segregación 1:1 (A.19).

2.7. Mapa Genético

La construcción de un mapa genético requiere el uso de un *software* especializado para poliploides o dedicado a genoma complejos como el de la caña de azúcar. La tarea es dar el orden correcto para los marcadores, por tanto, se evaluaron diferentes *softwares* para decidir cuál usar según la literatura y uso en caña. Cinco variables fueron tenidas en cuenta para analizar las herramientas computacionales encontradas: número de citas y usos, antigüedad (años), documentación, formato VCF y trabajos relacionados con caña. Estas variables fueron evaluadas con suma aritmética de sus valores enteros y cuando la variable era binaria "sí" o "no", su correspondiente numérico fue 1 o 0.

Las variables fueron definidas al usar una búsqueda indexada en Google Scholar. La variable Número de Citas y Usos fue calculada revisando los sitios web de cada herramienta y contando los artículos enlistados que la han citado, además, se usó el buscador con palabras clave como, por ejemplo, `OneMap software`, `OneMap package` o `OneMap program`. Para la variable de Antigüedad (años) se referenció el artículo inicial con el que se creó la herramienta hasta la fecha actual. Para la variable Documentación se revisó si el programa tenía un tutorial o repositorio en línea que explicara sus funcionalidades y elementos. Para la variable Formato VCF se buscó en la documentación si habían coincidencias con el patrón "vcf" que indicara que este formato es soportado por el programa. Para la última variable, Usado para Caña, se realizaron búsquedas con palabras clave: `OneMap software + sugarcane`, `OneMap package + sugarcane` o `OneMap program + sugarcane`. Para la primera variable (Número de Citas y Usos) y la última (Usado para Caña) se registraron las sumatorias de los resultados obtenidos en cada búsqueda (ver tablas A.4 y A.5). Dentro de las herramientas, la más destacada fue JoinMap v5.0 (JM) (Van Ooijen, 2018).

En JM se usaron los SNPs seleccionados y filtrados. JM recibe un formato de archivo como entrada para leer los marcadores SNPs. Este formato se logró usando la instrucción de NGSEP v3.3.3 `ConvertVCF` que convierte de un archivo en un formato `.vcf` a un formato `.txt` (ver algoritmo A.20). En el encabezado de este archivo (p. ej. `pop_jm.txt`) se indicaron 4 variables en diferentes renglones: nombre de la población (`name = CP_biparental`), el tipo de población (`popt = CP`), en este caso es *Cross pollinators* para todas las estrategias usadas, es decir, una población resultante del

cruce entre dos padres diploides heterogéneamente homocigotos y heterocigotos, según el manual de usuario de JM (Van Ooijen, 2018), finalmente, el número de loci (p. ej. `nloc = 3739`) y la cantidad de individuos (`nind = 95`). Después del encabezado con estas variables, el contenido restante del archivo son los loci con sus respectivos genotipos por individuo que ya estaban descritos en el archivo `.vcf` antes de ejecutar la instrucción `ConvertVCF`.

En las estrategias se fijó una frecuencia de recombinación (FR) máxima de 0.5 y un *Logarithm of Odds* independiente (indLOD) de ≥ 9.0 para evitar falsos ligamientos (Costa *et al.*, 2016; Gutierrez *et al.*, 2018). El tamaño del mapa genético en cM dividido el número de loci fueron usados para calcular la densidad `cM/loci`. El tamaño del mapa genético en kbp dividido el número de loci fueron empleados para calcular la densidad `kbp/loci`. Ambas densidades obtenidas fueron divididas para el cálculo del cociente, es decir, $\frac{\frac{cM}{loci}}{\frac{kbp}{loci}} = \frac{cM}{kbp}$. Estas densidades y el cociente son útiles para saber la relación que hay entre los centiMorgans (cM), kilo pares de bases (kbp, *kilo base pairs*) y marcadores (loci) en los mapas genéticos. En la estrategia A se usó un valor indLOD de 9.0 según lo sugerido en Balsalobre *et al.* (2017) y Garsmeur *et al.* (2018), una FR de 0.5 y se escogieron en JM los Grupos de Ligamiento (GL) con más de 20 loci. En la estrategia B se usó un indLOD de 15.0, una FR de 0.5 y se seleccionaron GL con más de 20 loci. En la estrategia C se usó nuevamente un indLOD de 9.0, pero esta vez se usó una FR de 0.015 y se escogieron GLs con al menos 20 loci.

Resultados

3.1. Secuenciación de una población F1 de caña de azúcar

Por cada individuo de la población F1 de caña de azúcar se obtuvieron dos archivos, el primero con un conjunto de lecturas *forward* y el segundo con un conjunto de lecturas *reverse*. Estos datos fueron almacenados en formato FASTQ con una compresión del software `gzip`. Las lecturas de cada individuo fueron evaluadas por su tamaño de almacenamiento. Un total de 2,694.19 GB (2.63 TB) ocupan los datos de toda la población en disco. La notación usada para distinguir cada conjunto de lecturas e individuos es *P2-id-lectura*, donde *P2* es la identificación de esta población en CENICAÑA, *id* es el identificador de cada individuo y *lectura* es el conjunto de lecturas 1 (*forward*) o 2 (*reverse*). En la figura 3.1 está el total de los 190 conjuntos de lecturas, de los 95 individuos, que no alcanzan a representarse totalmente en el eje horizontal. El eje vertical muestra el tamaño en disco que ocupa cada uno de los conjuntos de lecturas por individuo. En toda la población, el individuo que presentó menos tamaño en su conjunto de lecturas *forward* fue el P2-40 con 10.48 GB y el individuo que presentó mayor tamaño, en su conjunto de lecturas *reverse*, fue P2-11 con 20.91 GB (Figura 3.1).

El conjunto total de lecturas por individuo fueron cuantificadas para obtener un promedio, un máximo y un mínimo poblacional. En total se recibieron 38,142,703,858 lecturas *forward* y *reverse* en toda la población. Cada lectura tiene 150 bp, entonces hay un total de 5,721,405,578,700 bp ca. 5.72 Tbp secuenciadas en toda la población. La notación usada para distinguir cada conjunto de lecturas *forward* y *reverse* por individuo es *P2-id*, donde *P2* es la identificación de esta población en CENICAÑA y *id* es el identificador de cada individuo. El individuo P2-40 fue el que tuvo un mínimo de lecturas pareadas (*forward* y *reverse*) con 149,178,616. El individuo P2-11 fue el que tuvo un máximo de lecturas pareadas con 287,750,217. El promedio poblacional se ubicó en 200,751,000 lecturas pareadas en la población. En la figura 3.2 se puede ver la distribución del número de lecturas por individuo. El eje horizontal muestra el total de los 95 individuos que no alcanzan a representarse todos en el eje. El eje vertical muestra la cantidad de lecturas que fueron secuenciadas por individuo. Estas variaciones en los datos de secuenciación en el conjunto de muestras pueden ocurrir por artefactos en las lecturas: preparación de las muestras, generación del cluster o secuenciación (Sims *et al.*, 2014). Además, pueden haber diferencias en el tamaño genómico de las muestras dado que las lecturas pareadas tienden a tener una buena cobertura (Korbel *et al.*, 2007).

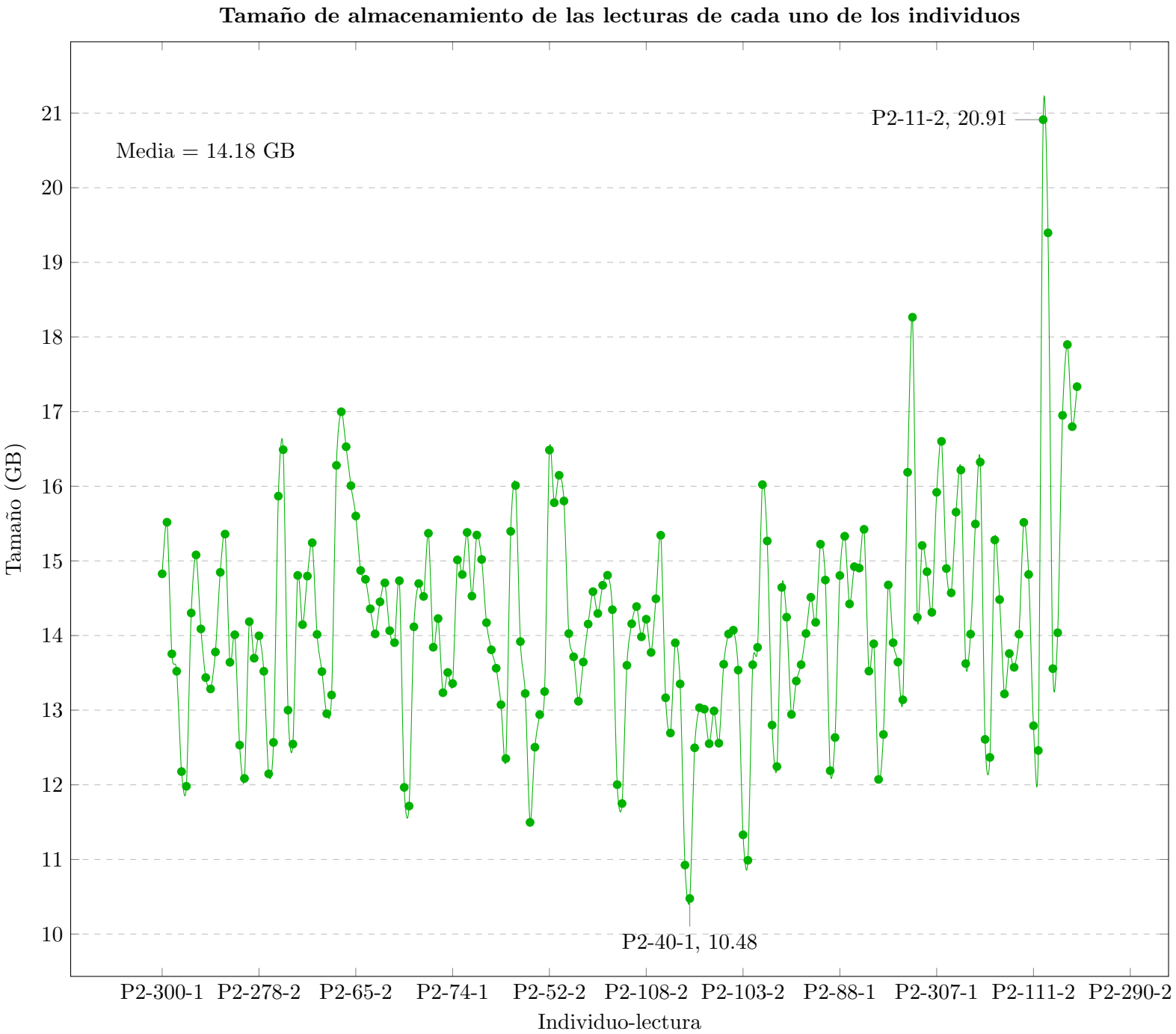


Figura 3.1: distribución de los tamaños en disco de las lecturas recibidas por cada individuo. El eje horizontal presenta los individuos (P2-id) con sus dos conjuntos de lecturas -1 (*forward*) y -2 (*reverse*).

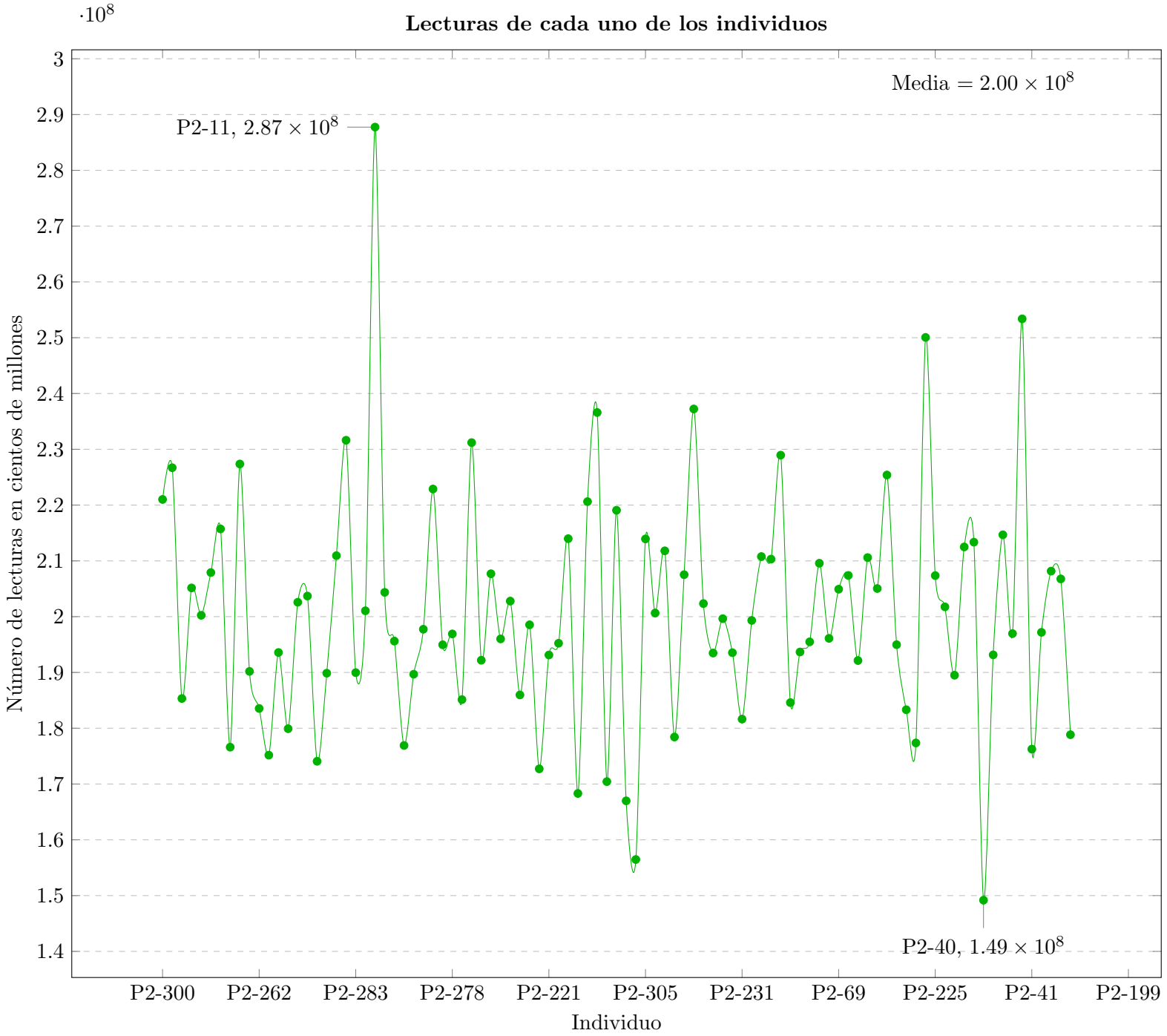


Figura 3.2: distribución del número de lecturas *paired-end* por individuo. A diferencia de la figura 3.1, aquí se muestra un consolidado de las lecturas *paired-end* (*forward* y *reverse*) por cada individuo.

3.2. Evaluación y análisis de las secuencias y alineamientos

El *software* FastQC v11.8 tiene 11 módulos para evaluar la calidad de las secuencias. Estos 11 módulos fueron evaluados para cada uno de los conjuntos de lecturas *forward* y *reverse* para cada individuo. El *software* entrega unos resultados uno a uno por cada par de conjuntos de lecturas para cada uno de los 95 individuos. No obstante, entrega un resumen para entender de manera general si el individuo-lectura tuvo éxito en el resultado del módulo (`PASS`), el resultado está dentro de lo esperado, pero podría ser mejor (`WARN`) o el resultado falló la prueba y debería corregirse (`FAIL`). De lo obtenido se deduce que la población cumplió las pruebas en un 95.7 % al considerar `PASS` y WARN como pruebas aceptables y `FAIL` como inaceptable (figura 3.3). En cada uno de los módulos graficados en la figura 3.3 hay un total de 190 conjuntos de lecturas, de los 95 individuos, que no alcanzan a representarse totalmente en el eje horizontal, y el eje vertical muestra tres clasificaciones según el resultado del módulo respectivo: `PASS`, `WARN` y `FAIL`.

En la figura 3.3f se puede observar un gran número de individuos que no pasaron la prueba. Este módulo hace referencia al contenido de Guanina y Citosina (GC). Esta prueba se hace para identificar si las lecturas tienen material contaminante pues se esperaría una distribución aproximadamente normal en el contenido de GC. Sin embargo, este resultado también puede ser evidencia de que hayan regiones en el genoma de la caña que tengan diferente contenido de GC. Para este caso, ambos eventos son probables, pero, más adelante, contrastan con el alto porcentaje de alineamiento de las lecturas a la referencia, por tanto, hay confianza para seguir con el análisis posterior.

Los resultados obtenidos con FastQC v11.8 del valor Phred fueron revisados por cada posición en la extensión del conjunto de lecturas. El valor Phred es usado para representar qué tan confiable es el asignamiento del llamado de una base por el proceso de secuenciación. Un valor alto indica una alta probabilidad de que el llamado a una base sea correcto, pero al contrario, un valor bajo indica una alta probabilidad de que el llamado a una base sea incorrecto. Es decir, este valor puede ser interpretado como un estimativo de error, qué tan probable es que una base sea llamada incorrectamente por el secuenciador, o como un estimativo de certeza, qué tan probable es que una base sea llamada correctamente. Por regla general, un valor Phred de 20 o mayor es aceptable porque significa que hay un 99 % de certeza con un 1 % de error (Ewin y Green, 1998).

El valor Phred estima la calidad de una secuencia, y para cada una de las muestras, esta fue mayor de 20. La calidad de la secuenciación se muestra por posición en el tamaño de las lecturas de 150 bp en la figura 3.4. La línea roja representa el valor de la mediana, unos cuadrados amarillos (no presentes aquí) representarían el rango intercuartil (25-75 %). Las extensiones de arriba y abajo son el 10 % y 90 %, respectivamente, y la línea azul representa la media (calidad) (figura 3.4). El eje Y muestra valores de calidad (valor Phred). Entre más alto este sea, mejor es la lectura de la base. El gráfico divide el eje Y en muy buena calidad (verde), calidad razonable (amarillo) y lecturas de mala calidad (roja). El eje X representa los pares de bases de las lecturas, este puede cubrir un rango de pares de bases (15-19, 30-34, ...) o ser discretos por unidad (1,2,3, ..., 150) (figura 3.4). La figura 3.4a muestra el conjunto de lecturas *reverse* del individuo P2-55, el cual alcanzó los valores mínimos Phred (34.58) en la población. La figura 3.4b es el conjunto de lecturas *forward* del individuo P2-7 con los valores máximos Phred alcanzados (36.41). El promedio de calidad en valor Phred de la población fue de 36.00, el cual significa una probabilidad de error de 1 en 4,000 llamados a una base por el secuenciador. Ver tabla 3.1 y tabla 3.2 (descripción completa de la población en Información Suplementaria tabla A.2).

Después de evaluar la calidad de las secuencias, los alineamientos mostraron porcentajes de alineamiento >86 % al genoma de la variedad CC 01-1940. Por ejemplo, el individuo P2-250 presentó el porcentaje más bajo de alineamiento, 86.2 %. El individuo P2-295 mostró el porcentaje más alto, 88.7 %, mientras que el promedio general en la población fue de 87.6 % (figura 3.5). Estos porcentajes

Tabla 3.1: Resumen del valor Phred de 95 individuos arrojado por FastQC v11.8

P2-id-lectura	Métrica	Valor
P2-55-2	Mínimo	34.58
P2-7-1	Máximo	36.41
	Promedio	36.00

Tabla 3.2: Equivalencias de error según el valor Phred y su correspondiente certeza (1-error).

Valor Phred	Error	Certeza (1 − Error)
10	1/10 = 0.1	0.9
20	1/100 = 0.01	0.99
30	1/1,000 = 0.001	0.999
36	1/4,000 = 0.00025	0.99975
40	1/10,000 = 0.0001	0.9999
50	1/100,000 = 0.00001	0.99999
60	1/1,000,000 = 0.000001	0.999999

dan cuatro evidencias: (1) los datos recibidos por NOVOGENE son datos de caña que alinean a la referencia, (2) los porcentajes de alineamiento son altos en comparación con experiencias pasadas usando técnicas como GBS (*Genotyping-by-sequencing*) (Elshire *et al.*, 2011) y RADSeq (*Restriction site associated DNA sequencing*) (Daveand y Blaxter, 2010; Miller *et al.*, 2007), al alinearlas al genoma de Sorgo (*Sorghum bicolor*), con valores de 29.9 % y 14.9 %, respectivamente (Trujillo, 2020), (3) estos resultados incrementan la probabilidad de éxito para identificar marcadores moleculares SNPs que sean usados en la construcción del mapa genético y (4), el recurso invertido para obtener los datos WGS de la población y el ensamblaje a nivel de *scaffolds* fue efectivo.

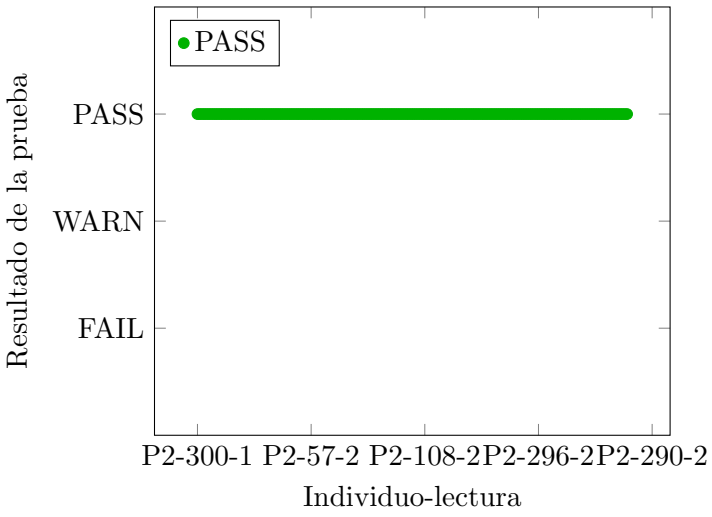
No obstante, en promedio hay un 12.4 % que no está alineando a la referencia y esto puede deberse por diferentes razones: (1) no continuidad en los *scaffolds* de la referencia entonces hay *gaps* o espacios vacíos en los cuales no puede haber alineación, (2) la referencia se trata de un genoma monoploide por lo que habrán lecturas que no estarán representadas en el ensamblaje y por tanto, no alinearán. El genoma monoploide no tiene todas las variantes por cada locus del genoma, en teoría, solo habrá una de las diez variantes posibles, y (3) puede haber contaminación en los datos y ese conjunto de datos no alinearán al ensamblaje. A pesar de lo anterior, los porcentajes altos de alineamiento dan certeza para usar las lecturas y seguir con el análisis posterior.

Dados los datos de alineamiento de los individuos, se pueden hacer otras pruebas de calidad en el *software* NGSEPv3.3.3: la de cobertura y la de calidad de la secuenciación en la lectura con respecto a la referencia. El análisis de cobertura indica, en promedio, cuántas veces una base en la referencia ha sido cubierta por las lecturas de un individuo, así, pueden haber bases que solo tengan una profundidad de una sola lectura o más de 300 lecturas. El análisis de calidad indica un porcentaje de pares de bases diferentes o *mismatches* que hay en un conjunto de lecturas de 150 bases con respecto a una referencia como los *scaffolds* de CC 01-1940.

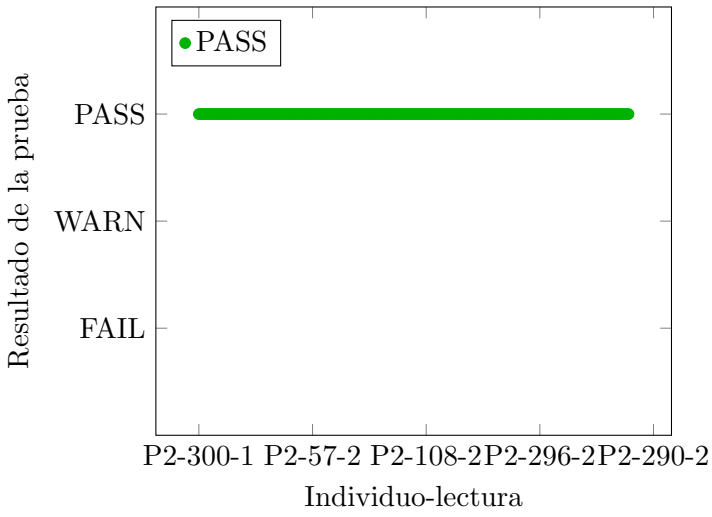
El análisis de cobertura arrojó al individuo P2-40 con un máximo que, a pesar de ser un gran número de pares de bases, que cubren diferentes posiciones en la referencia, solo alcanzó hasta 43X, lo que hace al individuo con menos profundidad de la población en su pico máximo (figura 3.6). Por el contrario, el individuo P2-11 tuvo su pico máximo de número de pares de bases cubiertas

a una profundidad de 88X, lo que lo hace el individuo con mayor profundidad de la población en su máximo (figura 3.6). Además, la cobertura promedio se calculó para toda la población, esto es promediar los resultados de cada uno de los individuos en cada nivel de profundidad y evaluar a qué nivel cae su máximo, para este caso, este fue de 56X (figura 3.6).

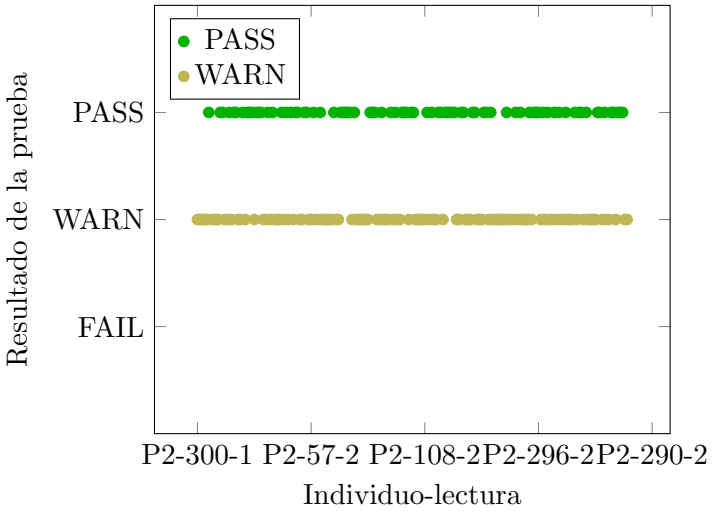
Lander y Waterman (1988) propusieron una teoría de redundancia de la cobertura (c): la profundidad media de cobertura de la secuenciación puede definirse teóricamente como $c = LN/G$, donde L es la longitud de lectura (150 bp), N es el número de lecturas (149,178,616 para el individuo P2-40) y G es la longitud del genoma haploide (en este caso, es el tamaño del genoma monoploide 1.019 Gbp (Trujillo, 2020), porque equivaldría al haploide en un sistema diploide). El resultado de este cálculo equivale a 22X, pero el promedio poblacional estuvo en 56X, más de dos veces de lo esperado. Por tanto, este comportamiento ofrece confianza para identificar si un marcador en un individuo es homocigoto al alelo de referencia, al alelo alternativo o heterocigoto.



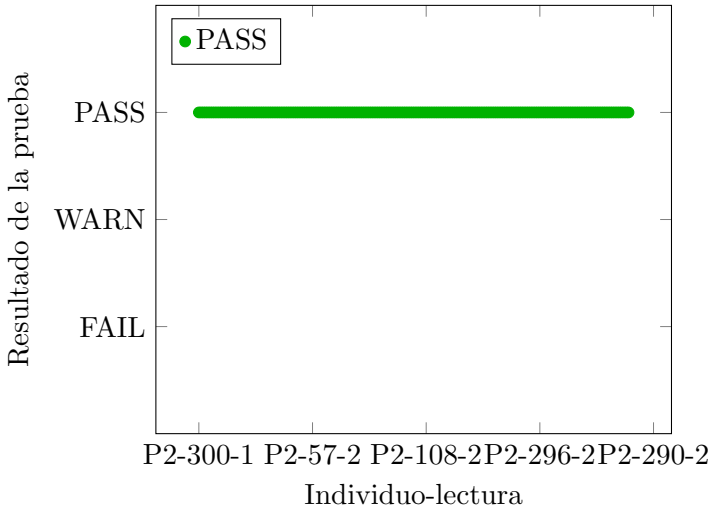
(a) Módulo *Basic Statistics*



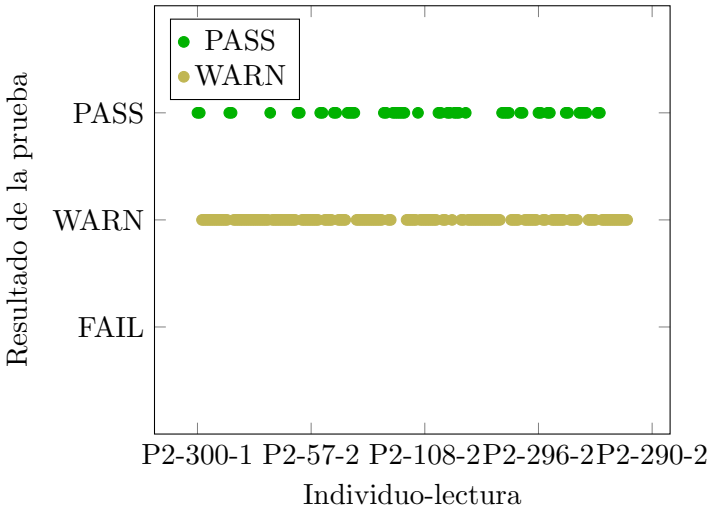
(b) Módulo *Per base sequence quality*



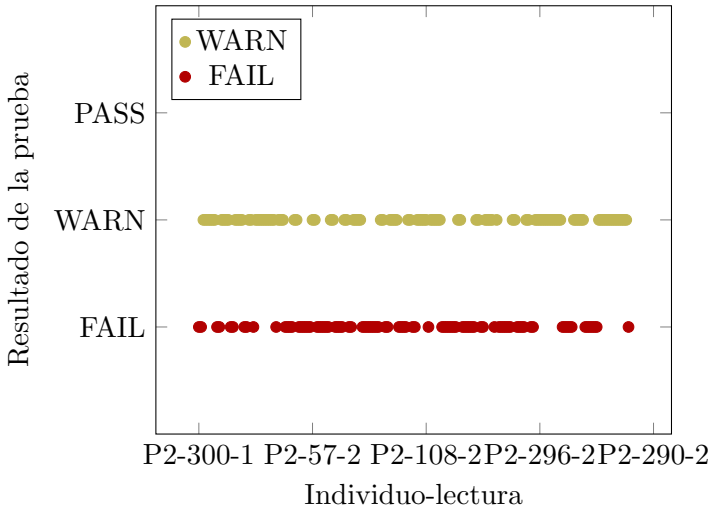
(c) Módulo *Per tile sequence quality*



(d) Módulo *Per sequence quality scores*

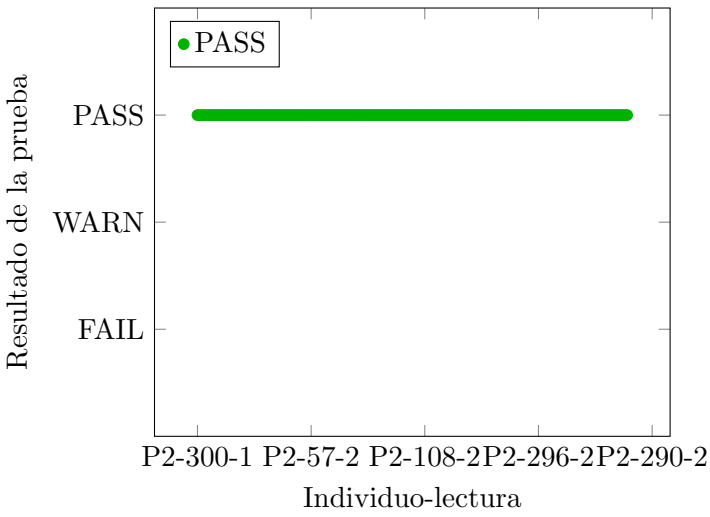


(e) Módulo *Per base sequence content*

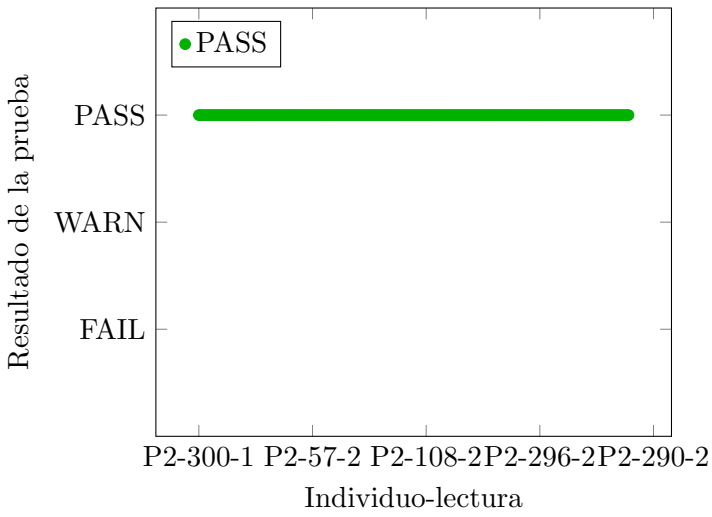


(f) Módulo *Per sequence GC content*

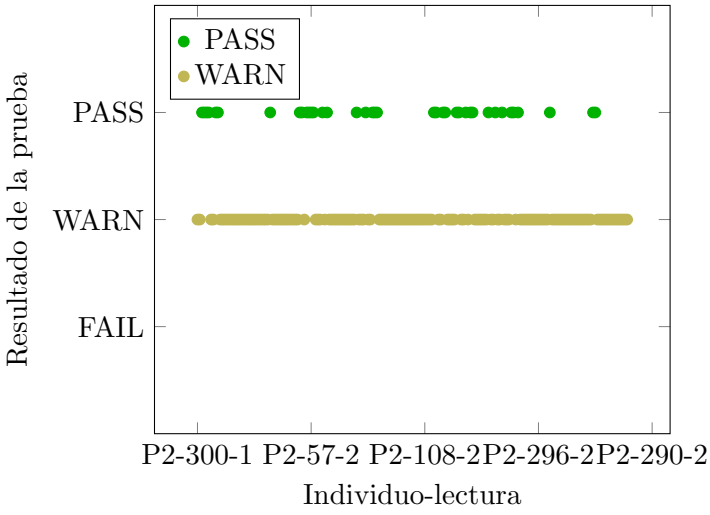
Figura 3.3: presentación de los primeros 6 (a-f) módulos que se evalúan con el *software* FastQC v11.8.



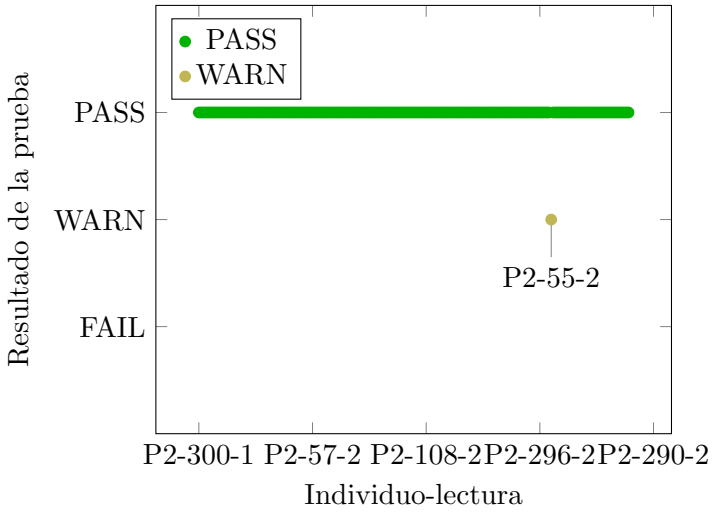
(g) Módulo *Per base N content*



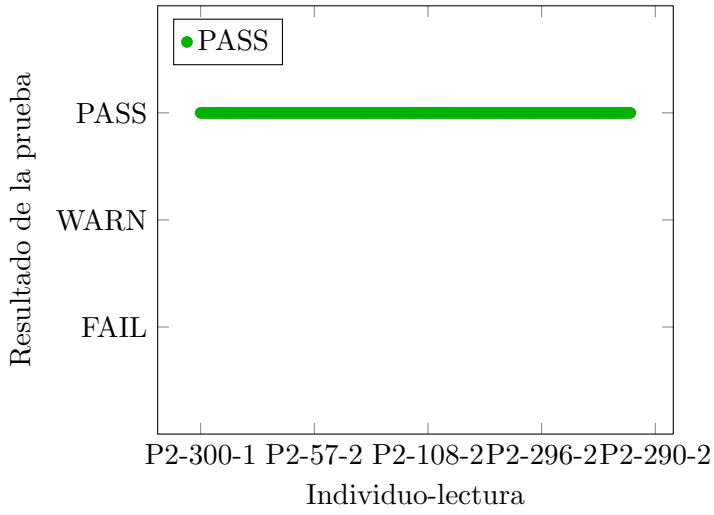
(h) Módulo *Sequence Length Distribution*



(i) Módulo *Sequence Duplication Levels*



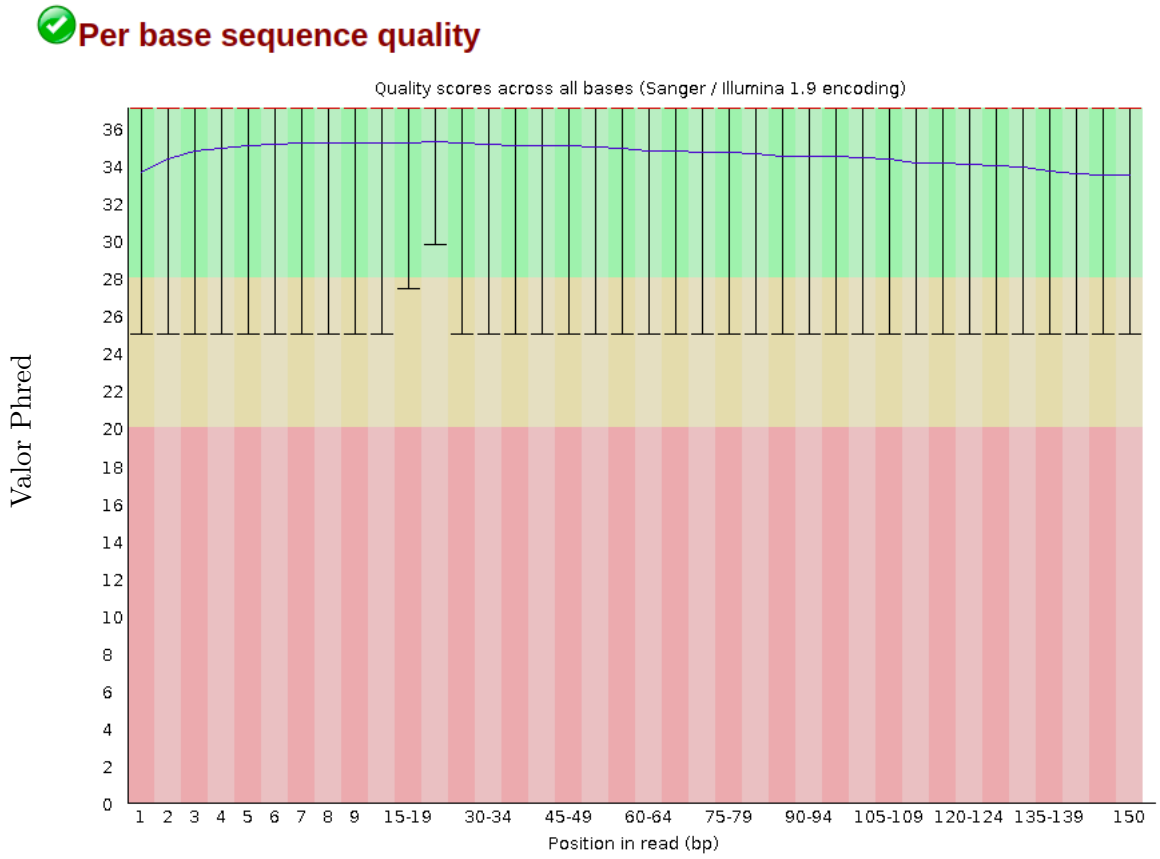
(j) Módulo *Overrepresented sequences*



(k) Módulo *Adapter Content*

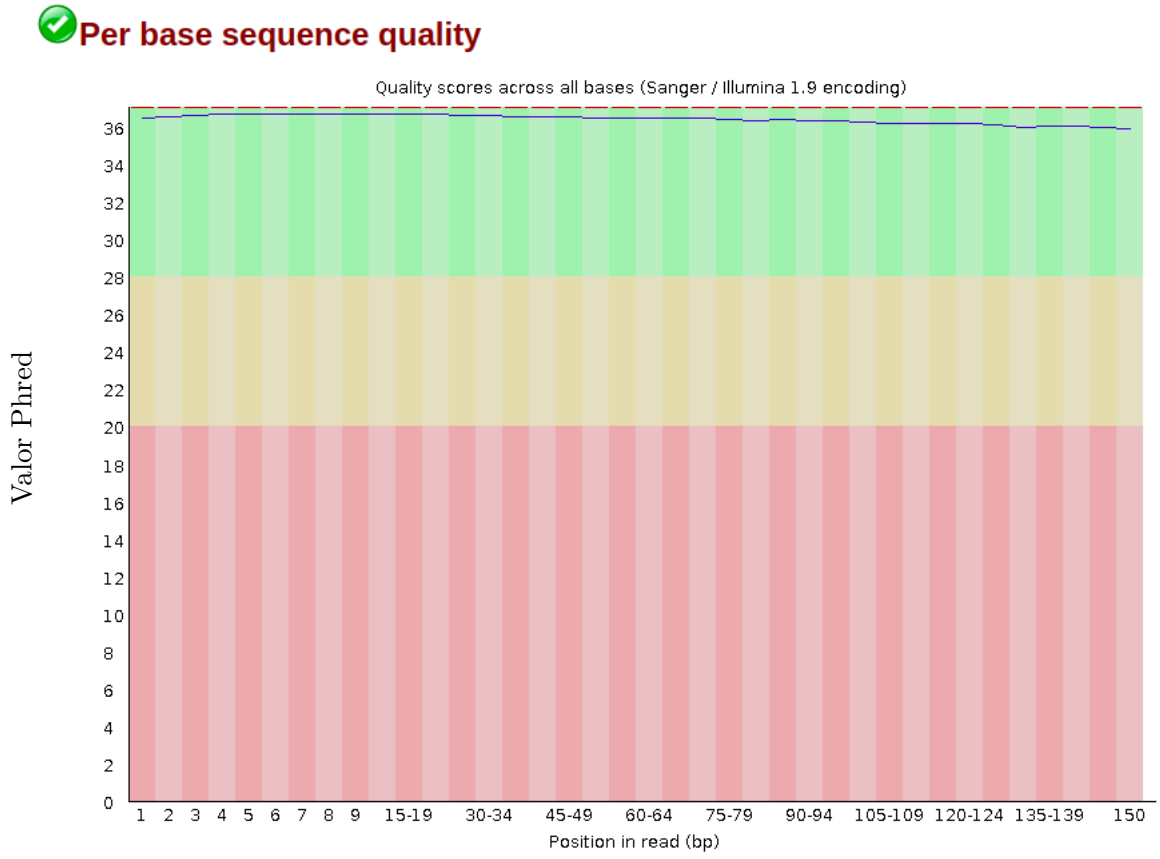
Figura 3.3: continuación de los últimos 5 (g-k) módulos que se evalúan con el *software* FastQC v11.8.

Calidad de la secuenciación en cada base del inviduo P2-55-2 (lecturas *reverse*)



(a) Posición en lectura (pb)

Calidad de la secuenciación en cada base del inviduo P2-7-1 (lecturas *forward*)



(b) Posición en lectura (pb)

Figura 3.4: gráfico de caja y extensión para los valores Phred en cada posición por conjunto de lecturas *forward* o *reverse*.

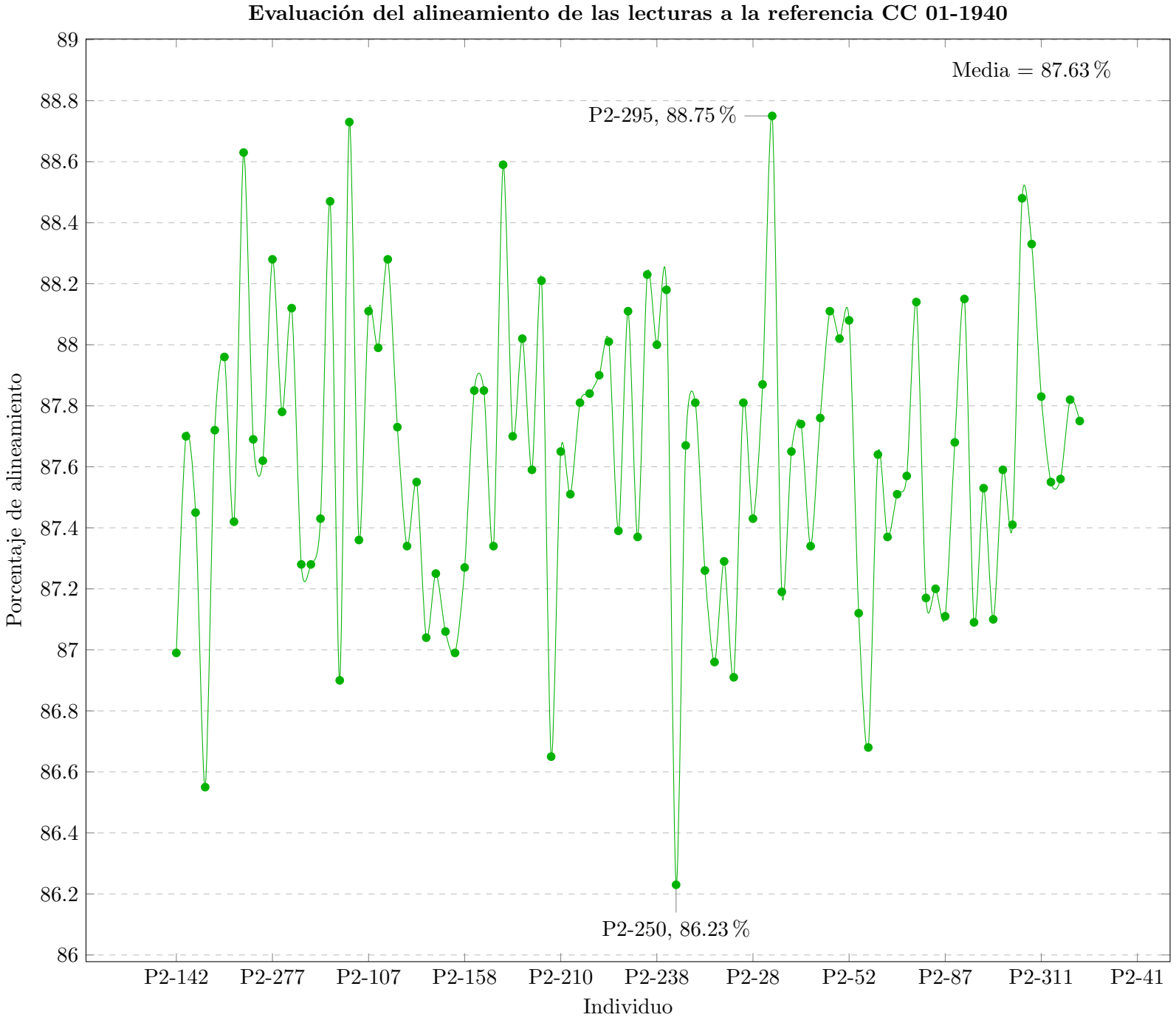


Figura 3.5: distribución del alineamiento de lecturas *paired-end* por individuo al genoma de referencia de CC 01-1940.

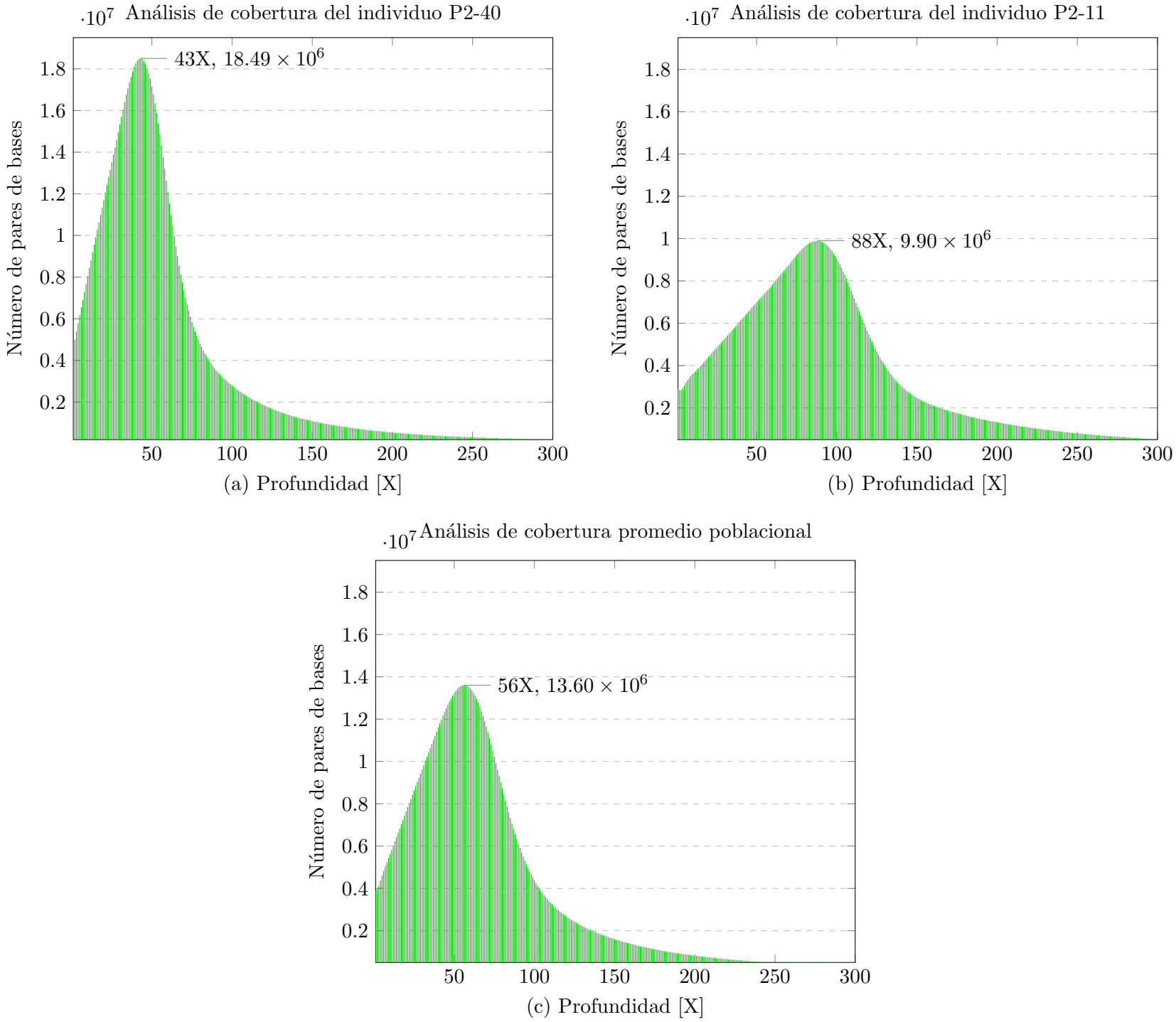
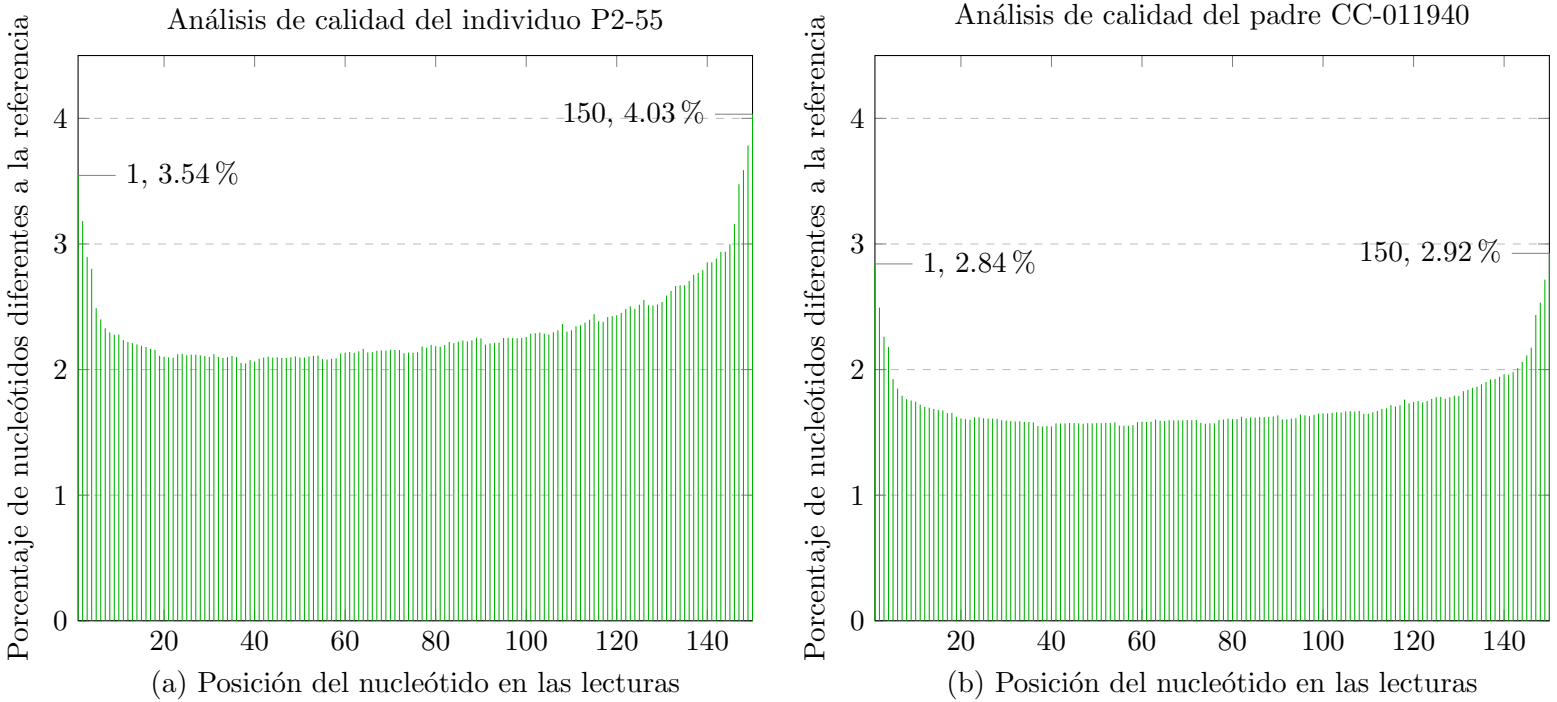


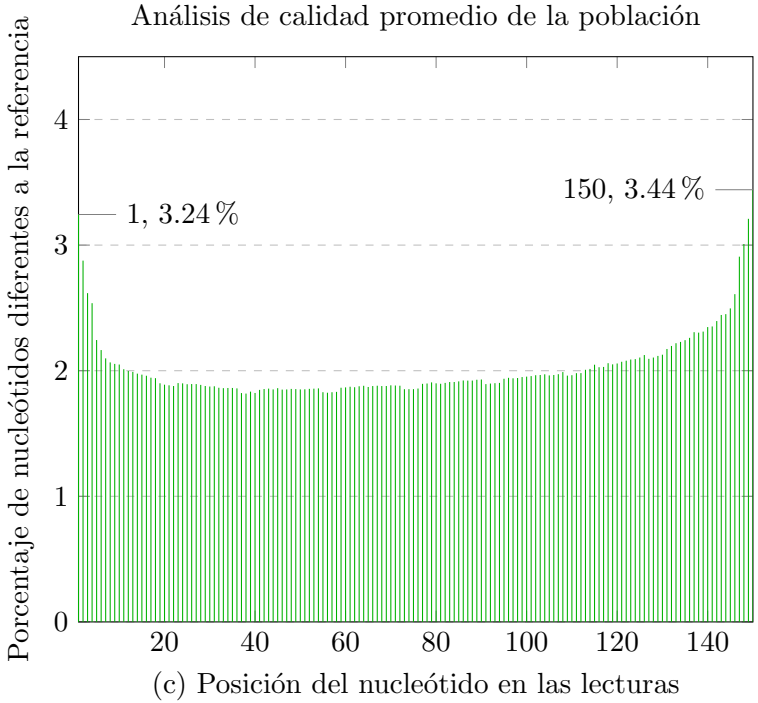
Figura 3.6: distribución del análisis de cobertura en la población hecho por el *software* NGSEPv3.3.3.

El análisis de calidad evidenció que el individuo P2-55, en la población, tuvo los mayores porcentajes de diferencia en pares de base con respecto a la referencia, en los diferentes alineamientos y desacoples (figura 3.7). El individuo CC 01-1940 (el padre), en la población, tuvo los menores porcentajes de diferencia en pares de base con respecto a la referencia. El genoma, al cual se alinea, es de este último individuo, por tanto, es de esperarse que la calidad de este fuera mayor que las demás muestras (figura 3.7). Así como en el análisis de cobertura, se produjo un gráfico del promedio poblacional, esto es promediar el porcentaje de bases diferentes a la referencia en cada posición en cada uno de los conjuntos de lecturas (150 bp) de los individuos (figura 3.7). En general, se puede ver en la figura 3.7 que las posiciones que tienen menor calidad son las de los extremos, pero eso está dentro de lo esperado porque el proceso de secuenciación es mucho más fidedigno en el centro de la secuencia (Meyer y Kircher, 2010). Los extremos son más suceptibles a *mismatches* debido a que son los sitios de ligamiento de los *primers* y puede haber ambigüedad en la lectura al emitirse la fluorescencia de uno u otro nucleótido (Lawrence *et al.*, 2017). El rango de porcentaje de nucleótidos diferentes con respecto a la referencia es aceptable para hacer el llamado de SNPs porque está dentro de lo esperado, en la literatura se ha reportado hasta un 4 % (Dohm *et al.*, 2008; Hoffmann *et al.*, 2009).



(a) Posición del nucleótido en las lecturas

(b) Posición del nucleótido en las lecturas



(c) Posición del nucleótido en las lecturas

Figura 3.7: distribución del análisis de calidad en la población utilizando la herramienta NGSEPv3.3.3.

3.3. Aplicación de filtros y llamado de variantes

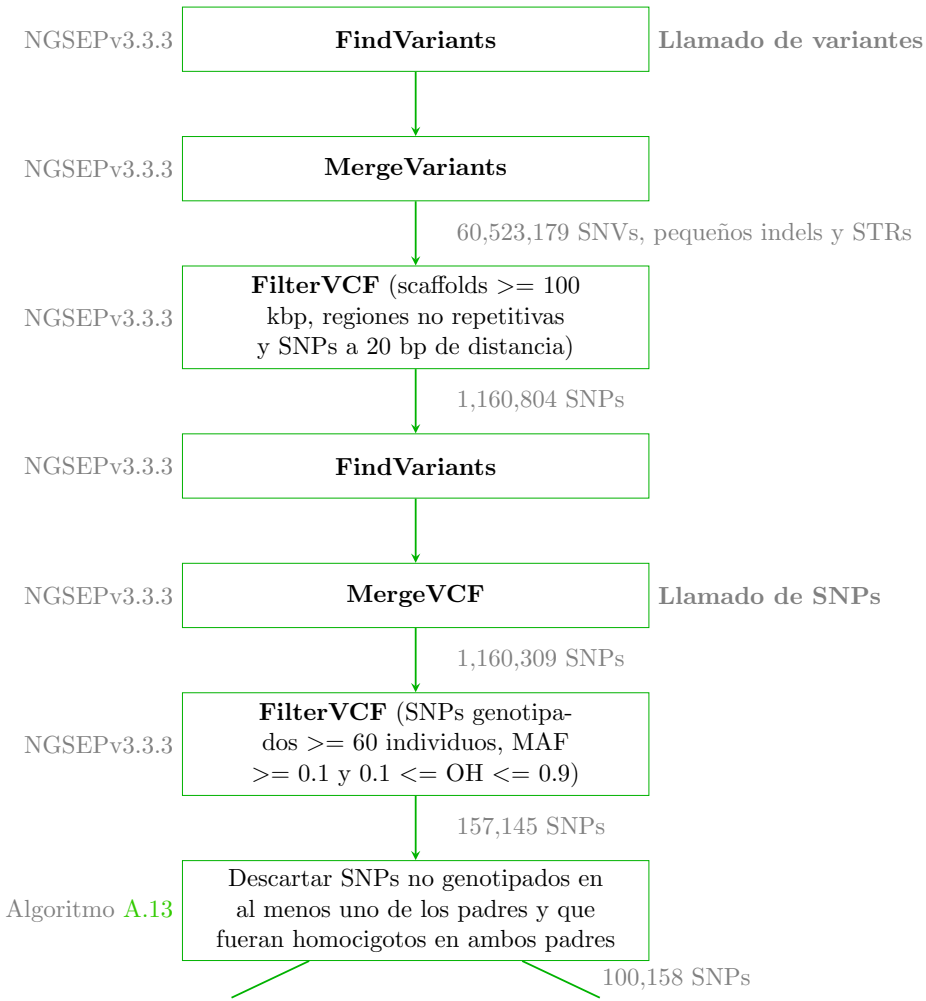
Los alineamientos obtenidos de cada individuo al genoma de referencia son la base necesaria y fundamental para hacer el *llamado de variantes*. Este llamado se hace necesario para la identificación de los marcadores moleculares SNPs y su objetivo es buscar inserciones, deleciones o mutaciones entre los individuos de la población, considerando los alineamientos y el genoma de referencia en una misma posición física. Este proceso se logró con NGSEP v3.3.3 con sus comandos `FindVariants`, `MergeVariants`, `FilterVCF` y `MergeVCF`. En cada una de estas instrucciones se tuvieron diferentes salidas según la estrategia usada (figura 3.8).

Con estas variantes seleccionadas, se hace un llamado de SNPs en el que se busca que los marcadores moleculares sean de calidad para asegurar la construcción y utilidad del mapa genético. Este proceso también se logró con NGSEP v3.3.3 y unos algoritmos adicionales construidos para este mismo propósito. En la figura 3.8 se puede ver un diagrama de flujo que incluye las etapas de llamado de variantes y de SNPs, o aplicación de filtros, y las estrategias A y B seguidas en el proceso de filtros. La estrategia C se presentan en la figura 3.9.

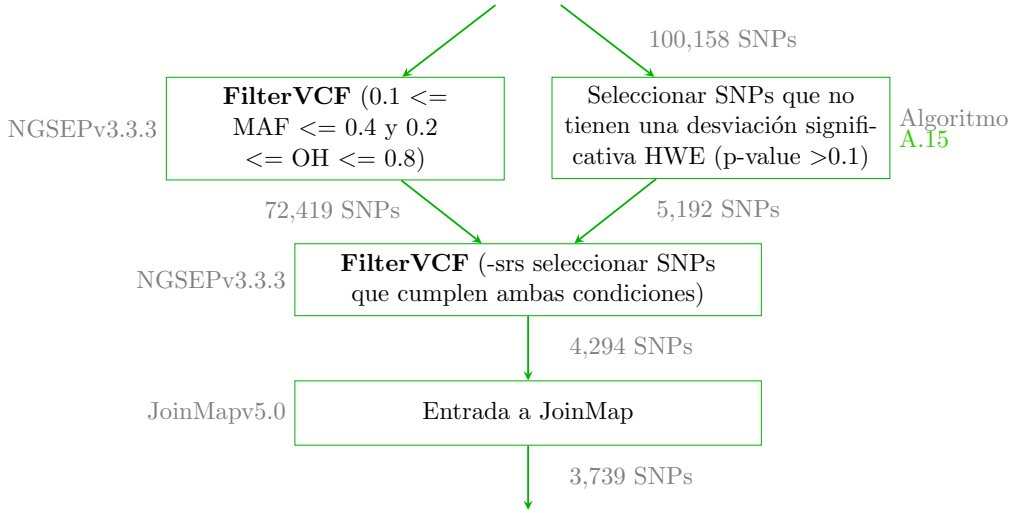
En las estrategias A y B, la lectura del número de loci en JoinMap v5.0 (JM) (Van Ooijen, 2018) fue reducida. En la primera estrategia, fueron 555 loci menos y, en la segunda, fueron 499 loci menos. Esta reducción se debió a inconsistencias en la genotipificación de los loci en algunos individuos de la población. La inconsistencia se presentaba cuando un individuo de la progenie tenía un genotipo, por ejemplo, homocigoto al alelo de referencia (0/0), pero un parental era heterocigoto (0/1) y el otro parental era homocigoto al alelo alternativo (1/1), lo cual no debería ser posible. Este último filtro, por procesamiento interno de JM, aseguró aún más que los loci fueran de calidad para la construcción del mapa genético. No obstante, los filtros, usados en la estrategia C, filtraron esas inconsistencias previamente a la entrada en JM y no hubo una disminución de los 7,816 SNPs de dicha estrategia (figura 3.9), lo que le confirió mayor cantidad de SNPs en comparación con las otras dos.

Dados estos filtros y el número de marcadores obtenidos, se procedió a graficar la heterocigosidad observada (OH) y la frecuencia del alelo menor (MAF) para ver cómo se comportaban ambas variables en la población para cada una de las estrategias. En la figura 3.10 está la distribución del OH y en la figura 3.11 está la distribución del MAF.

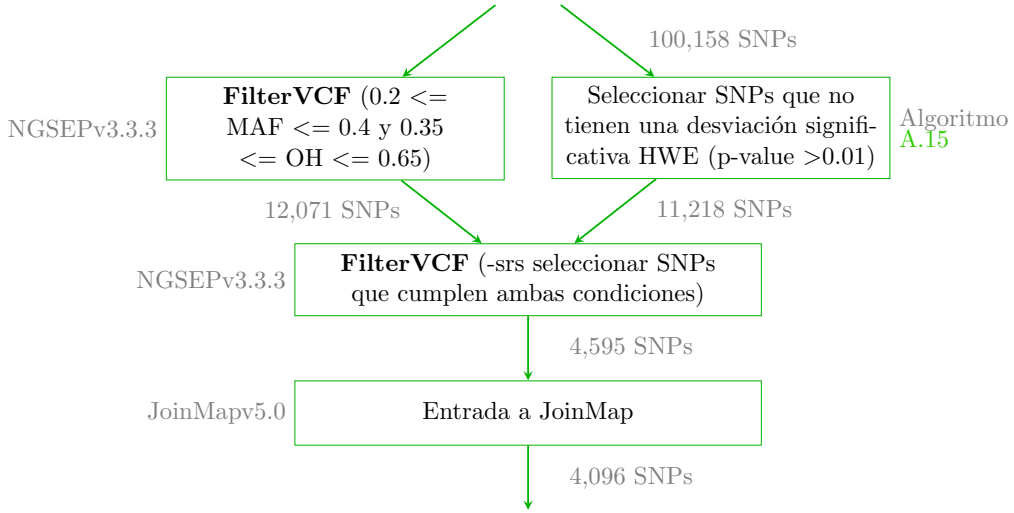
El concepto de la heterocigosis observada y la frecuencia del alelo menor fueron usados bajo la suposición de que la población es una F1, primera progenie resultado del cruce de dos parentales heterocigotos. Por consiguiente, las dosis alélicas deberían comportarse en la relación 1:2:1 . Según esta premisa y al considerar SNPs bialélicos, se establecieron los rangos de OH y MAF. El primero, para evitar altas heterocigosis o bajas heterocigosis porque pueden ser fenómenos debido a regiones repetitivas o que la mayoría de los SNPs sean homocigotos, lo cual no es normal en una población F1. El segundo, para evitar variantes poco comunes o alelos conservados en la población, este rango permite quedarse con variantes comunes, pero no conservadas. La OH y el MAF se usan porque ciertos loci pueden afectar erróneamente el análisis de ligamiento del mapa genético.



(a) Pasos iniciales para las estrategias A y B.



(b) Pasos exclusivos de la estrategia A.



(c) Pasos exclusivos de la estrategia B.

Figura 3.8: Conjunto de pasos seguidos como diagrama de flujo para hacer el llamado variantes y de SNPs para las estrategias A y B.

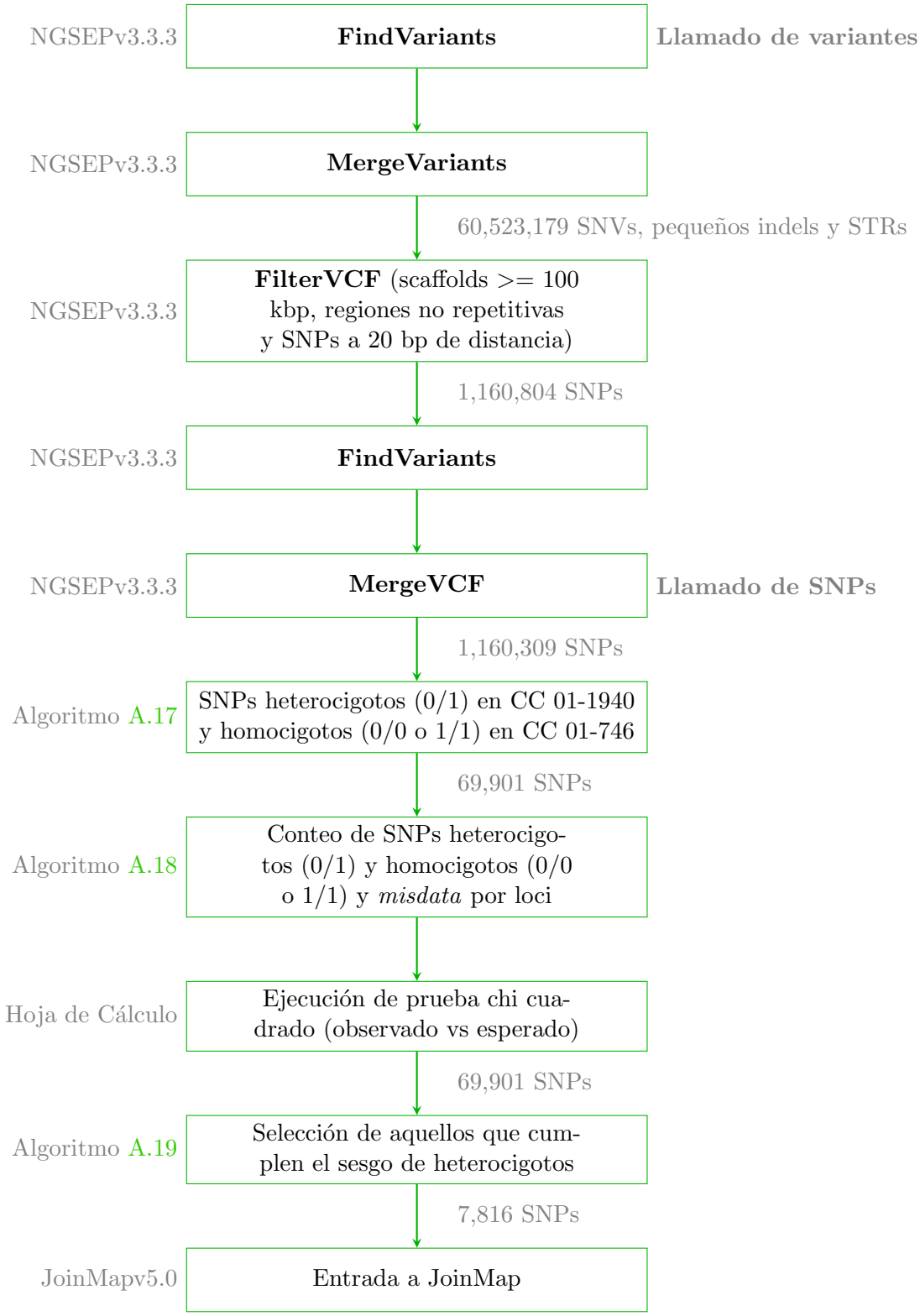


Figura 3.9: Conjunto de pasos seguidos como diagrama de flujo para hacer el llamado variantes y de SNPs para la estrategia C.

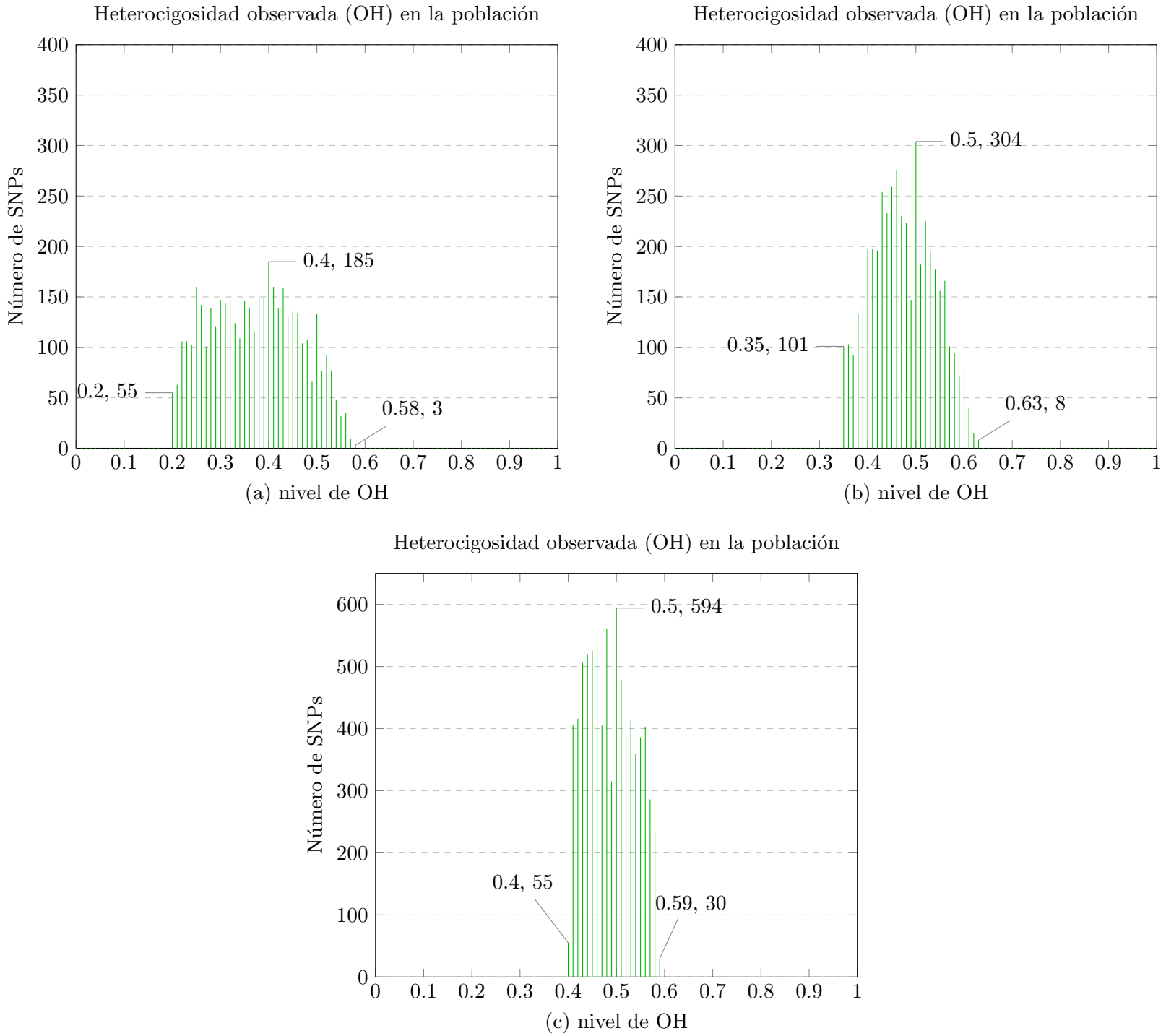
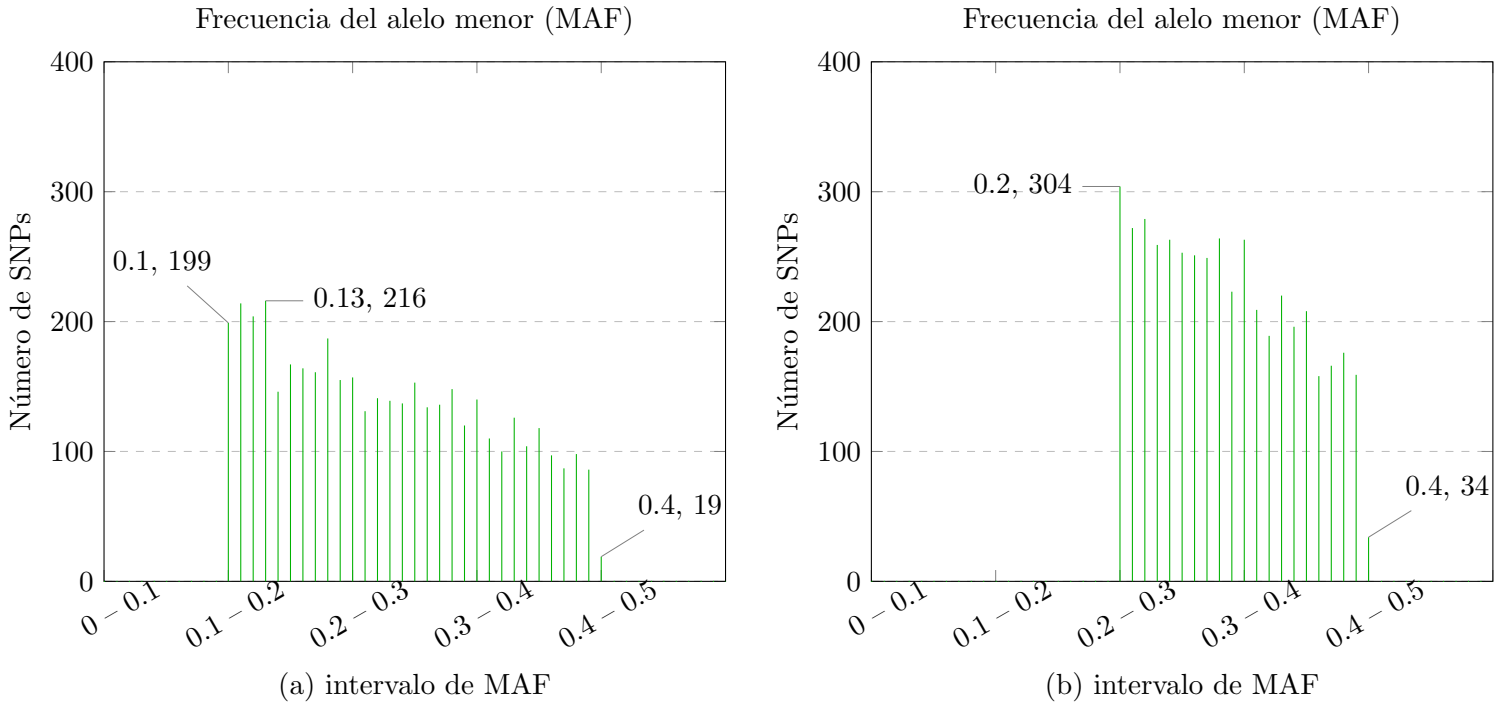


Figura 3.10: distribución del OH para los filtros aplicados a los SNPs de la población con la primera (a), segunda (b) y tercera (c) estrategia.



(a) intervalo de MAF

(b) intervalo de MAF

(c) intervalo de MAF

Figura 3.11: distribución del MAF para los filtros aplicados a los SNPs de la población con la primera (a), segunda (b) y tercera (c) estrategia.

3.4. Mapa genético

JoinMap v5.0 fue el *software* escogido para hacer el mapa genético por tener una puntuación total de 3,899 por encima del resto. OneMap v2.1.3 tuvo una puntuación total de 468, polymapR v1.1.0 tuvo 28 puntos y Netgwas v1.11 y Pergola v1.0 ambos tuvieron 12 puntos cada uno. La tabla 3.3 muestra en detalle los valores de cada parámetro considerado y la figura 3.12 muestra gráficamente la magnitud de estos valores.

Tabla 3.3: Parámetros considerados para evaluar el *software* más usado en caña y soportado por literatura científica para construir el mapa genético.

Softwares	**# Citas y usos**	**Antigüedad (años)**	**Documentación**	**Formato VCF**	**Trabajos relacionados con caña**	**Total**	**Referencia del *software***
JoinMap v5.0	2,760	27	Sí	Sí	1,110	3,899	Stam (1993)
polymapR v1.1.0	22	2	Sí	No	3	28	Bourke, van-Guesst *et al.* (2018)
Pergola v1.0	5	4	Sí	No	2	12	Grandke *et al.* (2017)
OneMap v2.1.3	349	13	Sí	Sí	104	468	Margarido *et al.* (2007)
Netgwas v1.11	6	3	Sí	No	2	12	Behrouzi *et al.* (2017)

3.4.1. Estrategia A

Esta estrategia de filtros con un valor indLOD de 9.0 y una FR de 0.1 (ver tabla A.6 para más experimentos) produjo un mapa genético con un tamaño de 9,679.50 cM y de 1,396,004 kbp con un total de 1,599 loci de un total de 3,739 loci iniciales que leyó JM. La tabla 3.4 presenta las estadísticas del mapa genético presentado en la figura 3.13. La fila de totales en la tabla 3.4 es igual a la sumatoria de las celdas anteriores a esta, exceptuando los parámetros de densidad y cociente. En este mapa genético no es mostrado el primer grupo de ligamiento porque JM no pudo graficarlo debido a su alto número de loci: 1,045.

3.4.2. Estrategia B

Esta estrategia de filtros con un valor indLOD de 15.0 y una FR de 0.5 (ver tabla A.7 para más experimentos) produjo un mapa genético con un tamaño de 1,191.30 cM y de 434,447 kbp con un total de 607 loci de un total de 4,096 loci iniciales que leyó JM. La tabla 3.5 presenta las estadísticas del mapa genético presentado en la figura 3.14. La fila de totales en la tabla 3.5 es igual a la sumatoria de las celdas anteriores a esta, con excepción de los parámetros de densidad y cociente.

3.4.3. Estrategia C

Esta estrategia de filtros con un valor indLOD de 9.0 y una FR de 0.015 (ver tabla A.8 para más experimentos) produjo un mapa genético con un tamaño de 1,198.74 cM y de 503,138 kbp con un total de 547 loci de un total de 7,816 loci iniciales que leyó JM. La tabla 3.6 presenta las estadísticas del mapa genético presentado en la figura 3.15. La fila de totales en la tabla 3.6 es igual a la sumatoria de las celdas anteriores a esta, con excepción de los parámetros de densidad y cociente.

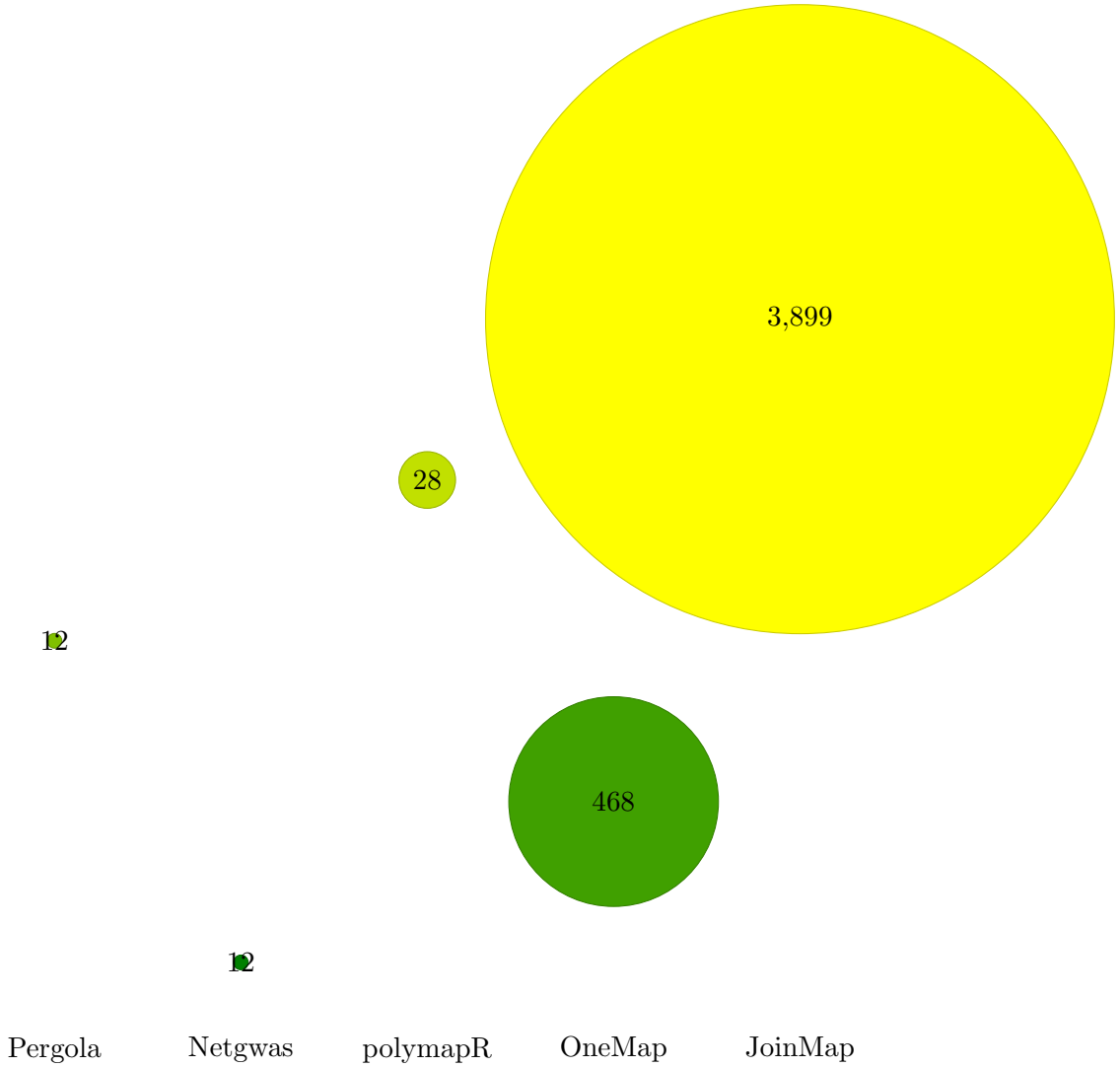


Figura 3.12: representación gráfica en un diagrama de burbujas del peso o valor que tiene cada *software* según los parámetros empleados.

Tabla 3.4: Estadísticas del mapa genético creado con la estrategia A.

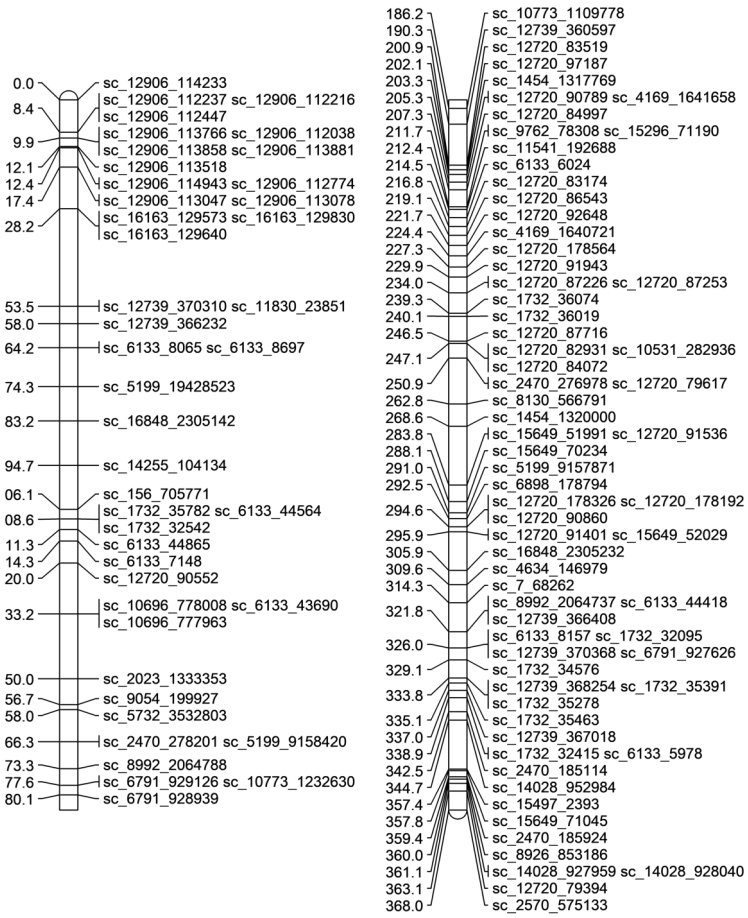
Grupo de ligamiento	Tamaño (cM)	Tamaño (kbp)	Loci	Densidad de marcador (cM/loci)	Densidad de marcador (kbp/loci)	Cociente (kbp/cM)
1	7,826.16	627,418	1,045	7.49	600.4	80.16
2	368.01	94,571	111	3.32	852	256.63
3	101.46	54,240	42	2.42	1,291.43	533.65
4	126.54	26,289	39	3.24	674.09	208.05
5	104.89	37,387	39	2.69	958.66	356.38
6	138.39	50,509	32	4.32	1,578.41	365.37
7	144.16	55,654	30	4.81	1,855.14	385.68
8	73.59	83,088	28	2.63	2,967.46	1,128.31
9	134.29	48,276	27	4.97	1,788.03	359.76
10	121.87	60,429	27	4.51	2,238.12	496.26
11	76.56	45,303	27	2.84	1,677.90	590.81
12	97.32	18,355	26	3.74	705.96	188.76
13	30.96	47,937	23	1.35	2,084.24	1,543.88
14	70.92	12,448	22	3.22	565.85	175.73
15	52.53	13,169	21	2.5	627.13	250.85
16	77.79	36,235	20	3.89	1,811.79	465.76
17	56.32	63,203	20	2.82	3,160.15	1,120.62
18	77.73	21,485	20	3.89	1,074.26	276.16
Total	**9,679.50**	**1,396,004**	**1,599.00**	**6.05**	**873.05**	**144.22**
Media	537.75	77,555.79	88.83	3.59	1,472.83	487.93
Desviación estándar	1,820.37	139,047.31	239.53	1.35	797.82	389.87

Tabla 3.5: Estadísticas del mapa genético creado con la estrategia B.

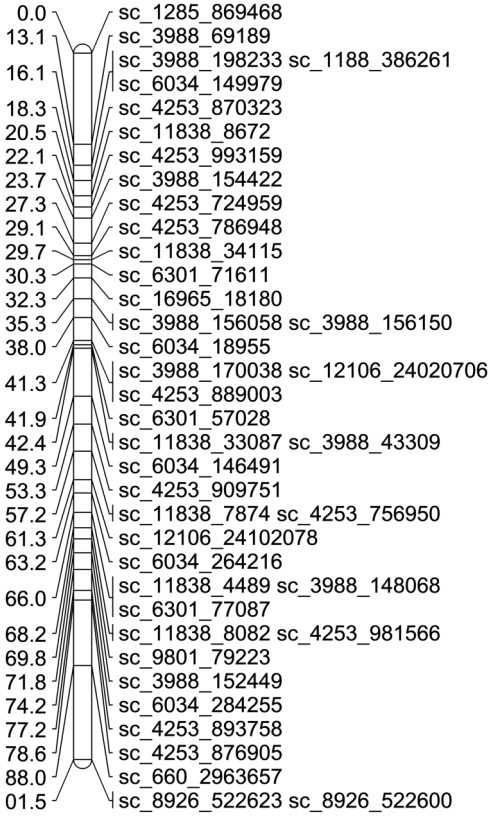
Grupo de ligamiento	Tamaño (cM)	Tamaño (kbp)	Loci	Densidad de marcador (cM/loci)	Densidad de marcador (kbp/loci)	Cociente (kbp/cM)
1	143.20	47,310	101	1.42	468.42	330.38
2	149.30	3,874	84	1.78	46.12	25.95
3	64.90	51,220	50	1.30	1,024.40	789.21
4	68.90	47,870	47	1.47	1,018.51	694.78
5	82.60	18,130	42	1.97	431.67	219.49
6	104.90	31,090	40	2.62	777.25	296.38
7	100.90	33,150	40	2.52	828.75	328.54
8	71.20	26,220	39	1.83	672.31	368.26
9	78.30	46,250	36	2.18	1,284.72	590.68
10	104.20	16,890	35	2.98	482.57	162.09
11	61.10	59,050	30	2.04	1,968.33	966.45
12	67.00	3,773	23	2.91	164.04	56.31
13	47.90	25,190	20	2.40	1,259.50	525.89
14	46.90	24,430	20	2.35	1,221.50	520.90
Total	**1,191.30**	**434,447**	**607**	**1.96**	**715.73**	**364.68**
Media	85.09	31,031.93	43	2.12	832.01	419.66
Desviación estándar	31.82	17,376.09	23	0.54	512.11	275.90

Tabla 3.6: Estadísticas del mapa genético creado con la estrategia C.

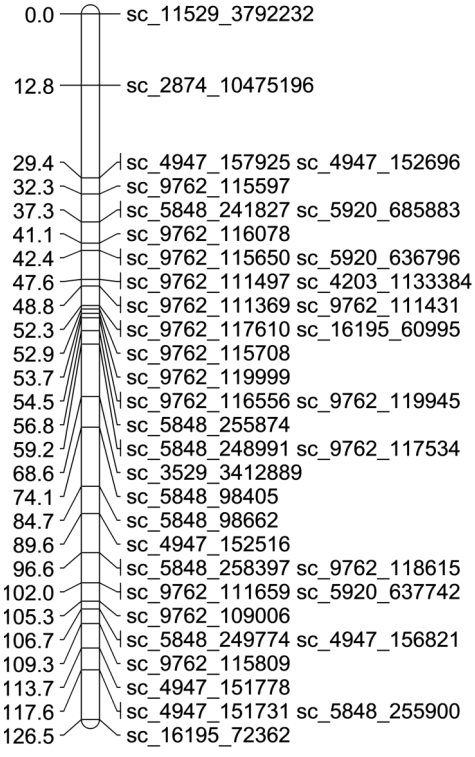
Grupo de ligamiento	Tamaño (cM)	Tamaño (kbp)	Loci	Densidad de marcador (cM/loci)	Densidad de marcador (kbp/loci)	Cociente (kbp/cM)
1	272.58	93,840	119	2.29	788.57	344.27
2	268.14	171,600	115	2.33	1,492.17	639.96
3	124.77	54,740	58	2.15	943.79	438.71
4	96.98	8,507	58	1.67	146.67	87.72
5	101.77	20,300	47	2.17	431.91	199.48
6	88.88	38,290	30	2.96	1,276.33	430.81
7	85.93	1,071	28	3.07	38.25	12.46
8	53.80	48,330	25	2.15	1,933.20	898.34
9	10.57	24,150	24	0.44	1,006.25	2,284.55
10	49.84	17,680	23	2.17	768.70	354.77
11	45.48	24,630	20	2.27	1,231.50	541.51
Total	**1,198.74**	**503,138**	**547**	**2.19**	**919.81**	**249.60**
Media	108.98	45,739.82	49.73	2.15	914.30	566.60
Desviación estándar	85.87	49,015.81	35.97	0.69	569.17	621.91



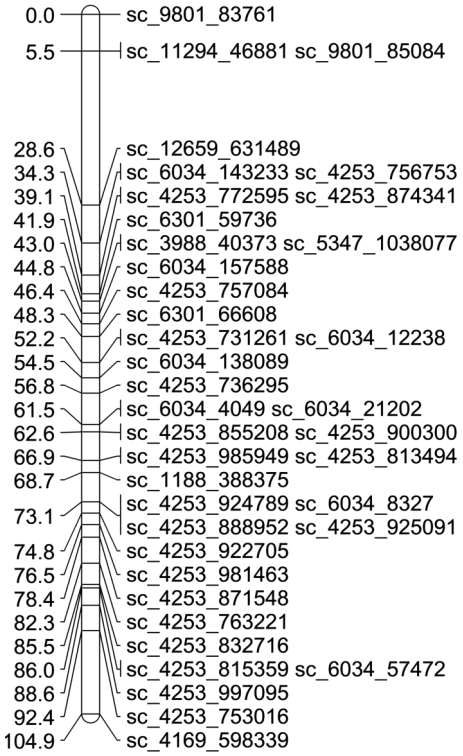
(a) 2do grupo de ligamiento con 111 loci.



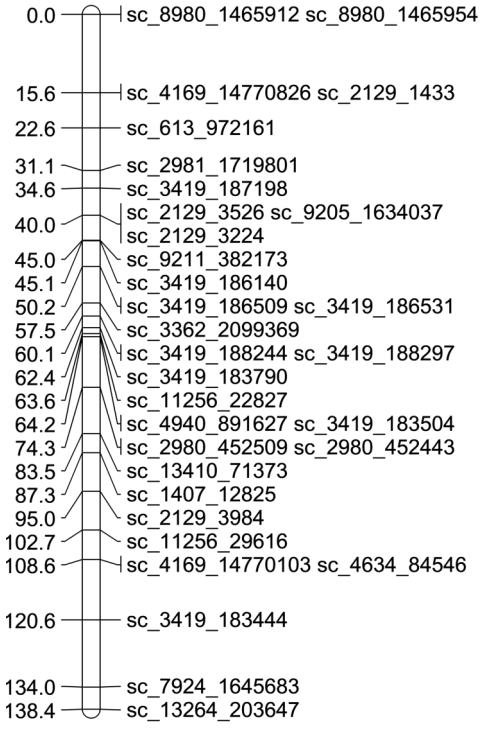
(b) 3er grupo de ligamiento con 42 loci.



(c) 4to grupo de ligamiento con 39 loci.

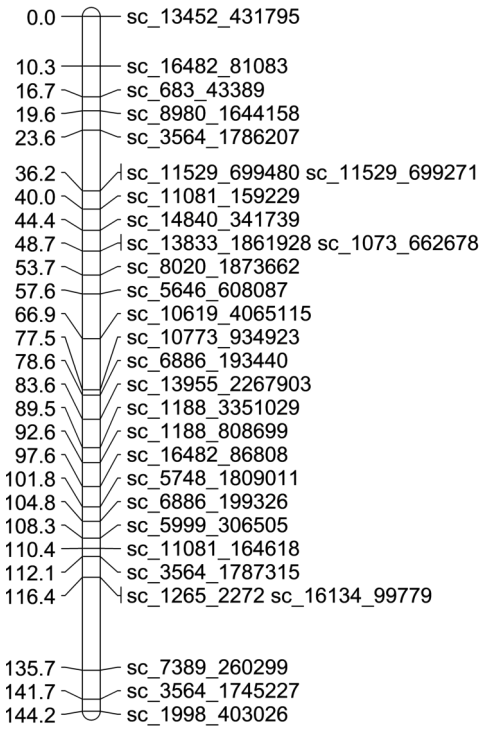


(d) 5to grupo de ligamiento con 39 loci.

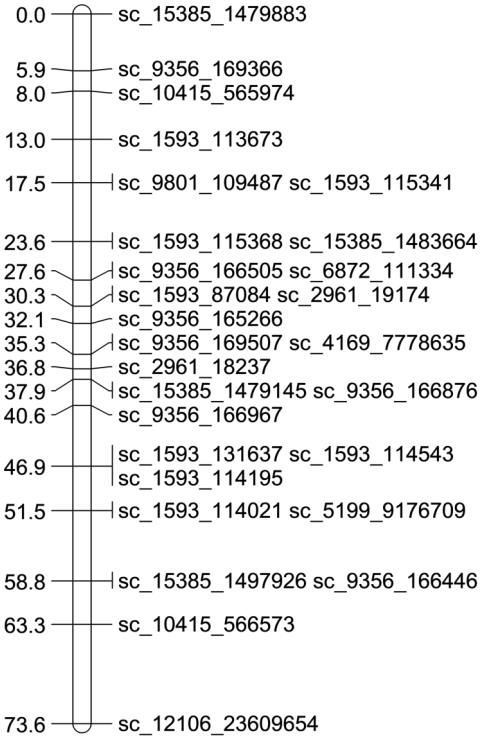


(e) 6to grupo de ligamiento con 32 loci.

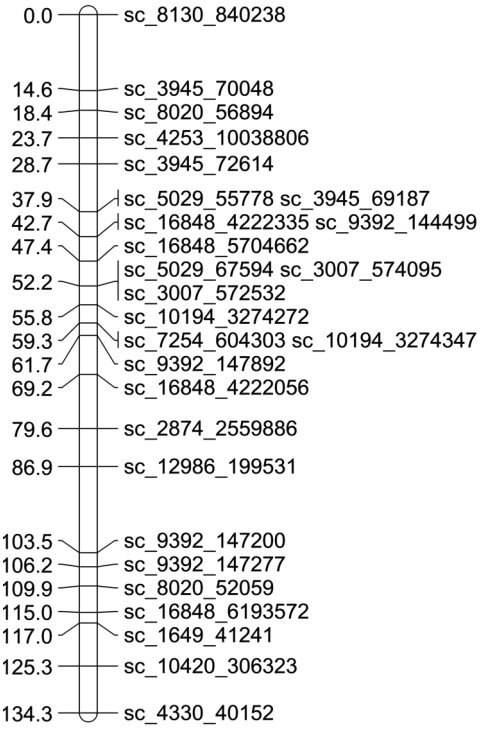
Figura 3.13: grupos de ligamiento 2 al 6 del mapa genético generado con JM. Estrategia A con un indLOD de 9.0.



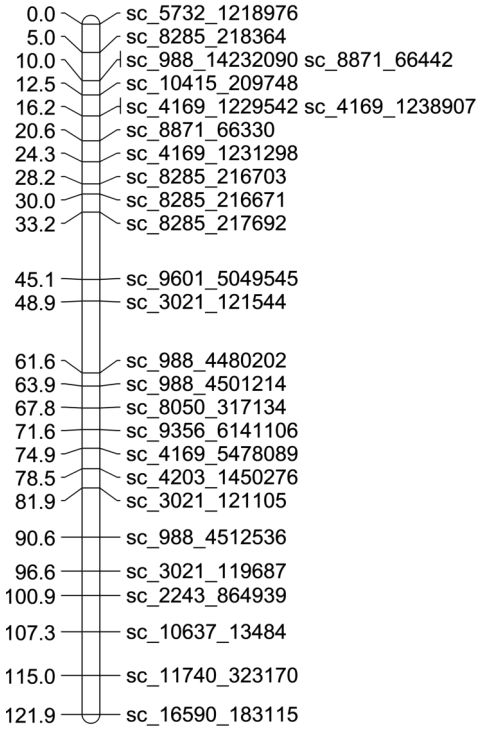
(f) 7mo grupo de ligamiento con 30 loci.



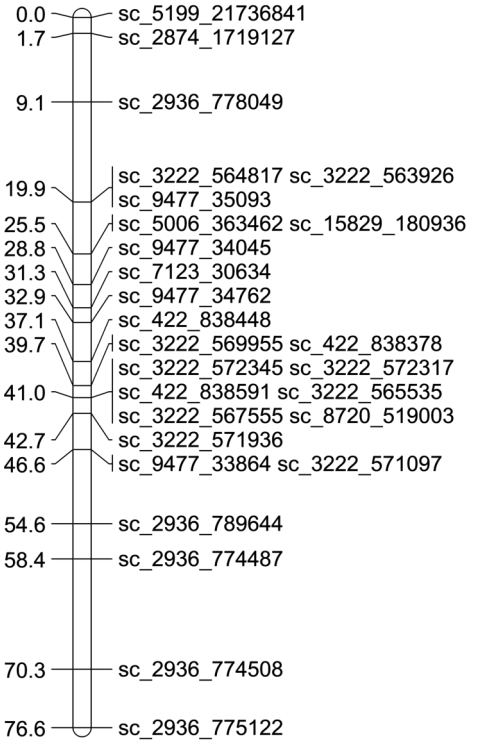
(g) 8vo grupo de ligamiento con 28 loci.



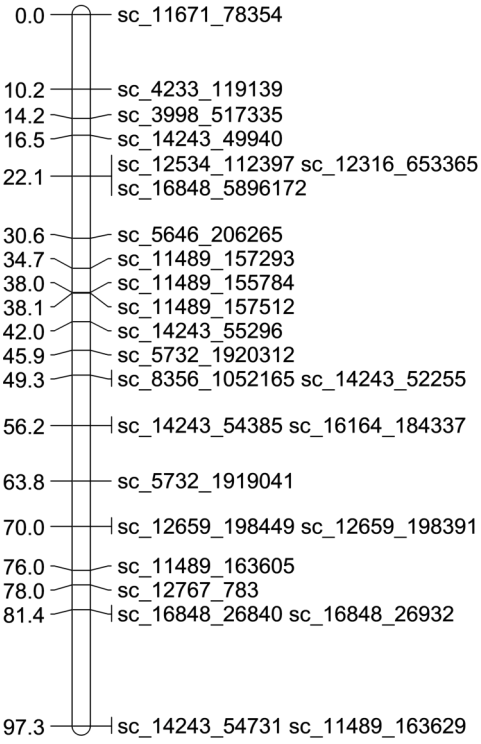
(h) 9no grupo de ligamiento con 27 loci.



(i) 10mo grupo de ligamiento con 27 loci.

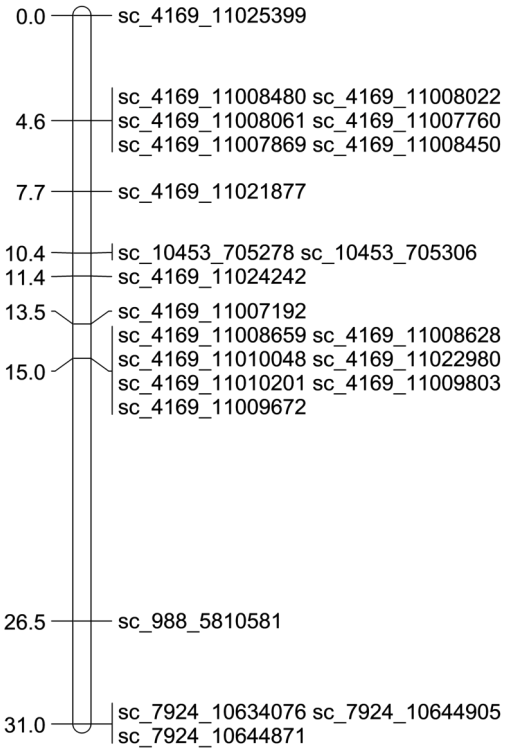


(j) 11ro grupo de ligamiento con 27 loci.

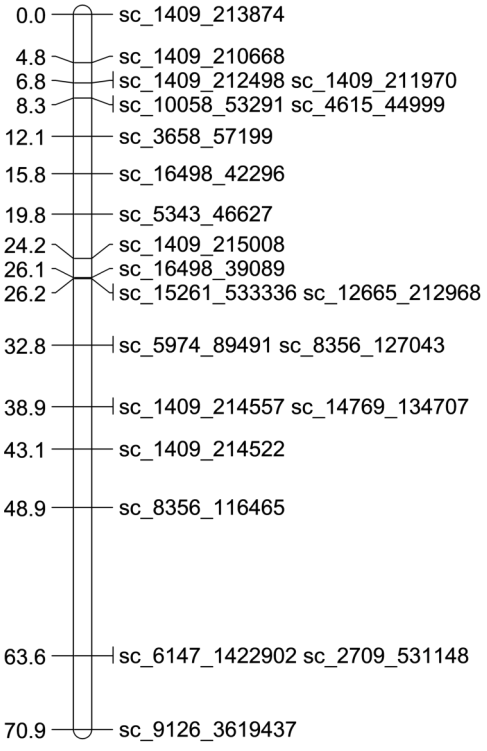


(k) 12do grupo de ligamiento con 26 loci.

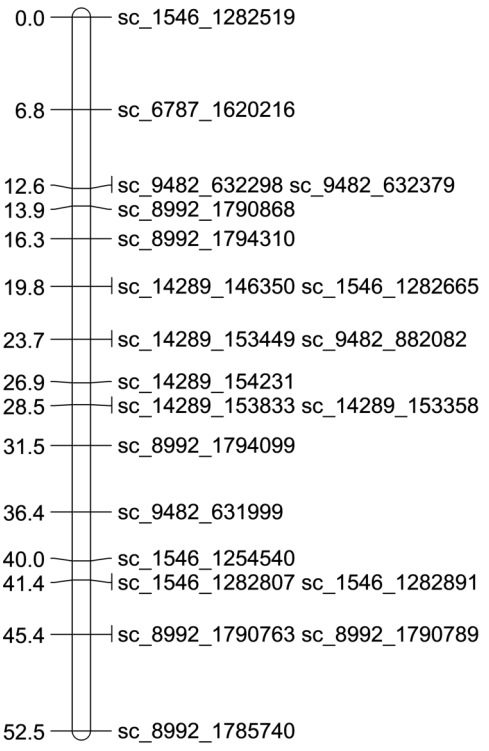
Figura 3.13: grupos de ligamiento 7 a 12 del mapa genético generado con JM. Estrategia A con un indLOD de 9.0



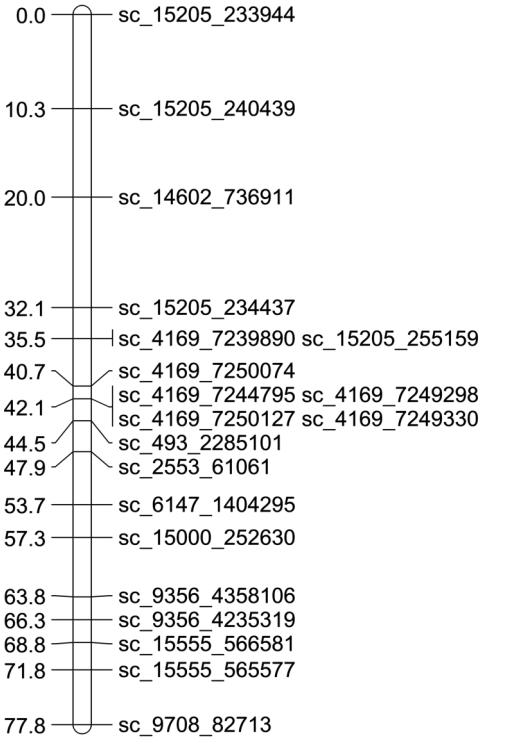
(l) 13ro grupo de ligamiento con 23 loci.



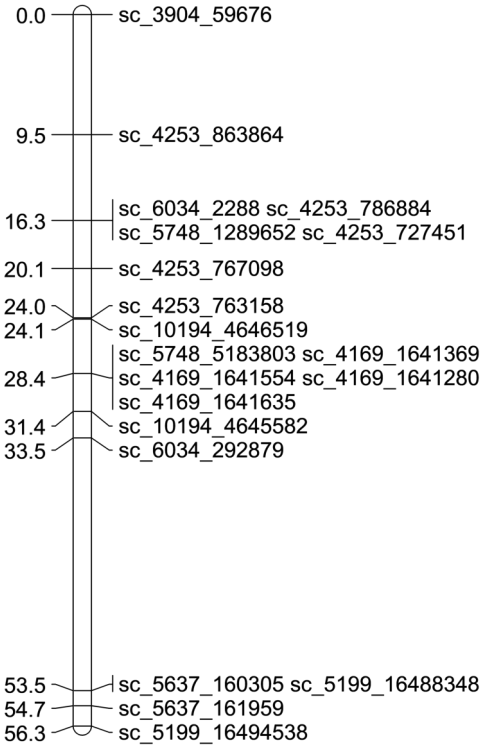
(m) 14to grupo de ligamiento con 22 loci.



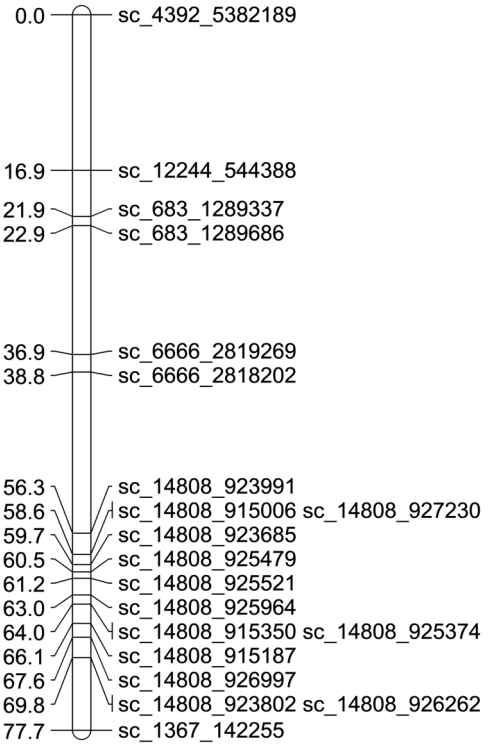
(n) 15to grupo de ligamiento con 21 loci.



(ñ) 16to grupo de ligamiento con 20 loci.

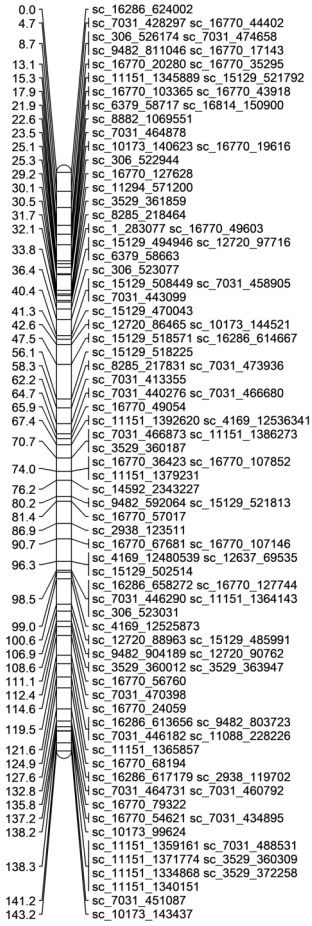


(o) 17mo grupo de ligamiento con 20 loci.

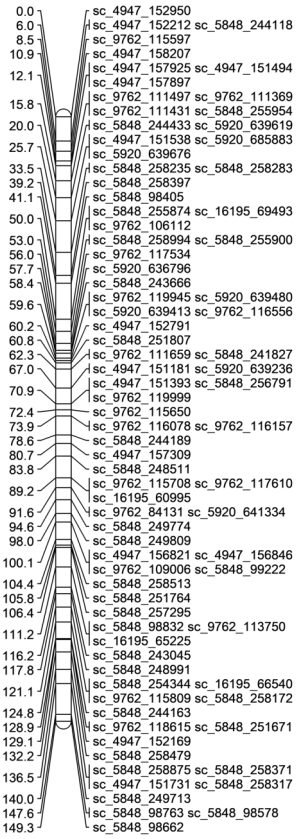


(p) 18vo grupo de ligamiento con 20 loci.

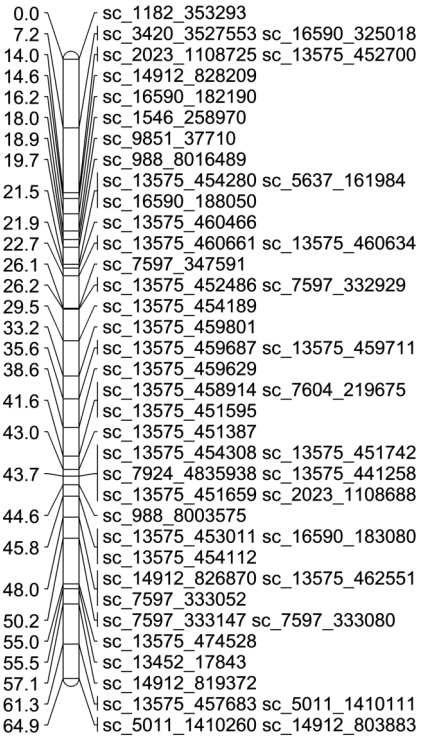
Figura 3.13: grupos de ligamiento 13 a 18 del mapa genético generado con JM. Estrategia A con un indLOD de 9.0



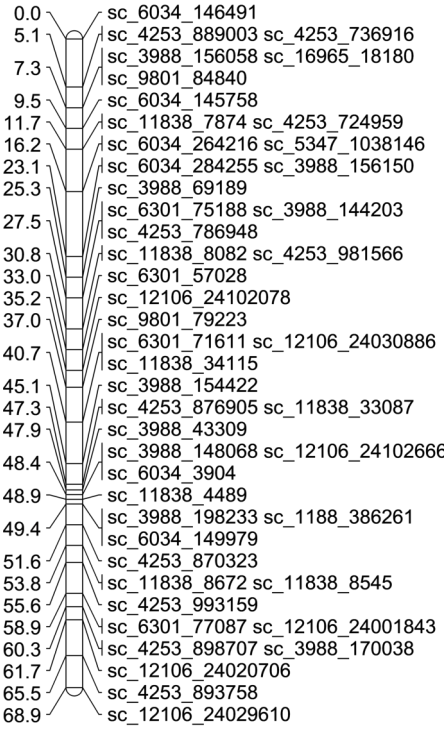
(a) 1er grupo de ligamiento con 101 loci.



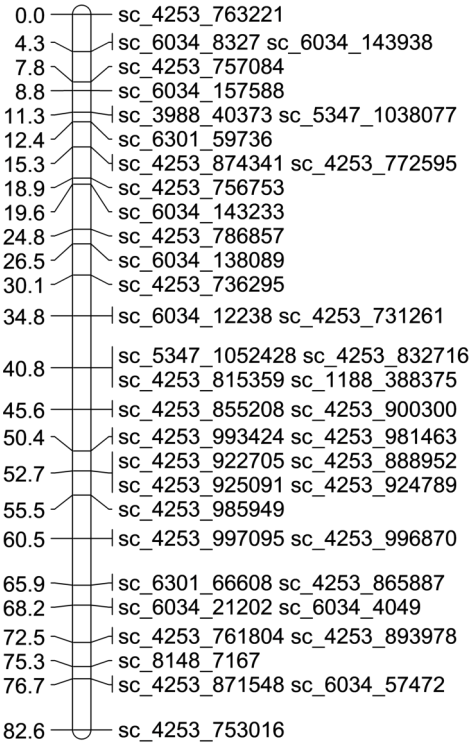
(b) 2do grupo de ligamiento con 84 loci.



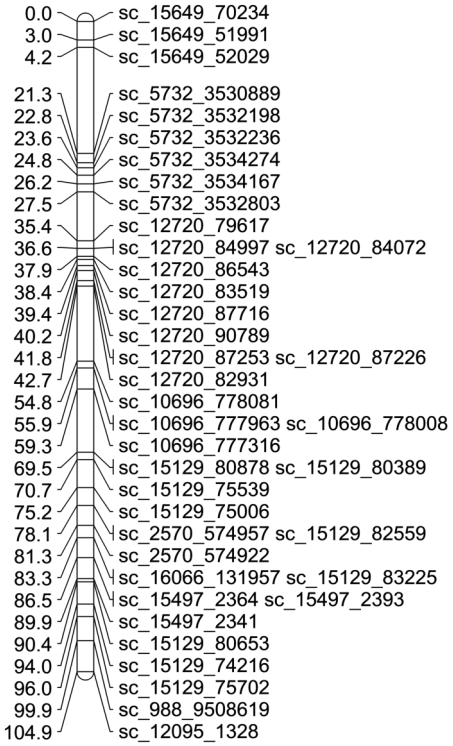
(c) 3er grupo de ligamiento con 50 loci.



(d) 4to grupo de ligamiento con 47 loci.

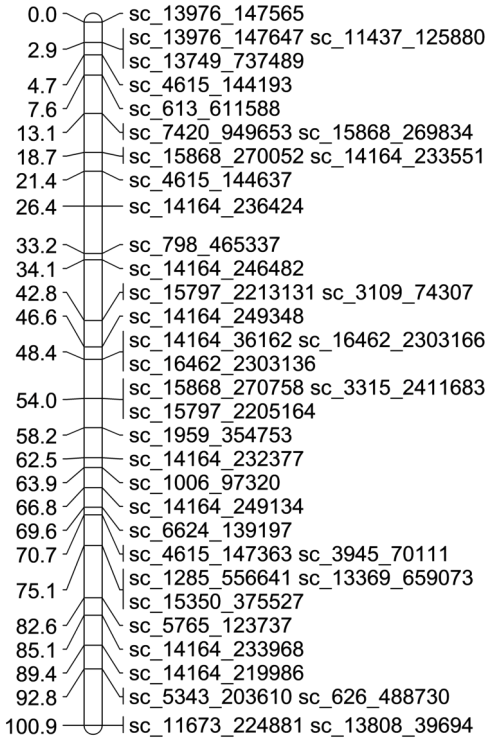


(e) 5to grupo de ligamiento con 42 loci.

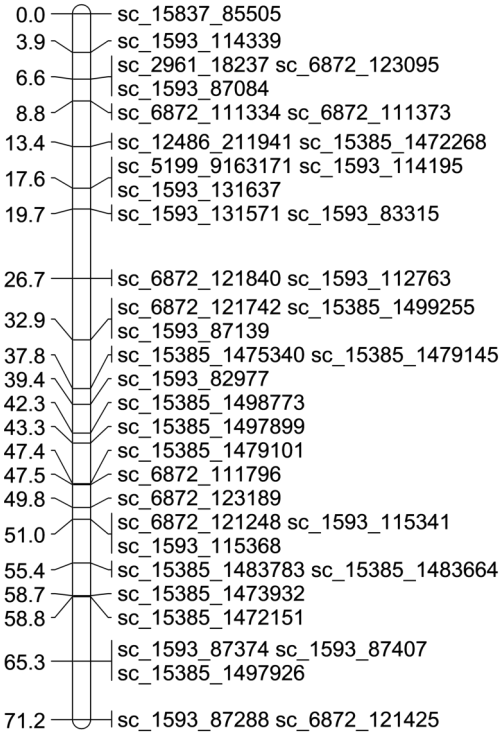


(f) 6to grupo de ligamiento con 40 loci.

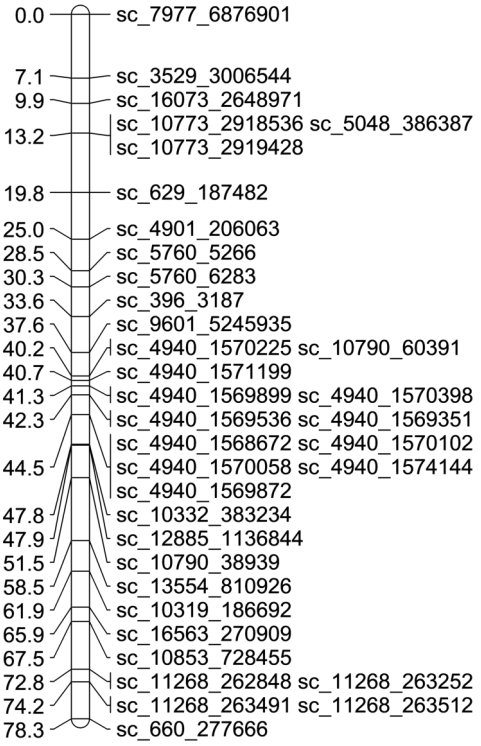
Figura 3.14: grupos de ligamiento 1 al 6 del mapa genético generado con JM. Estrategia B con un indLOD de 15.0



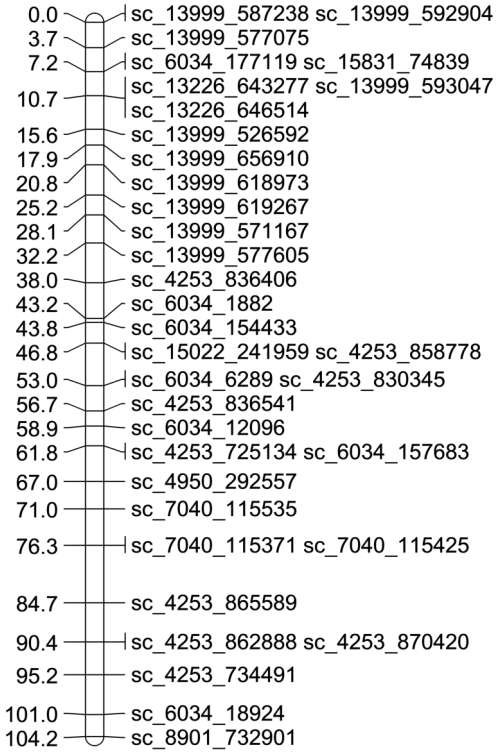
(g) 7mo grupo de ligamiento con 40 loci.



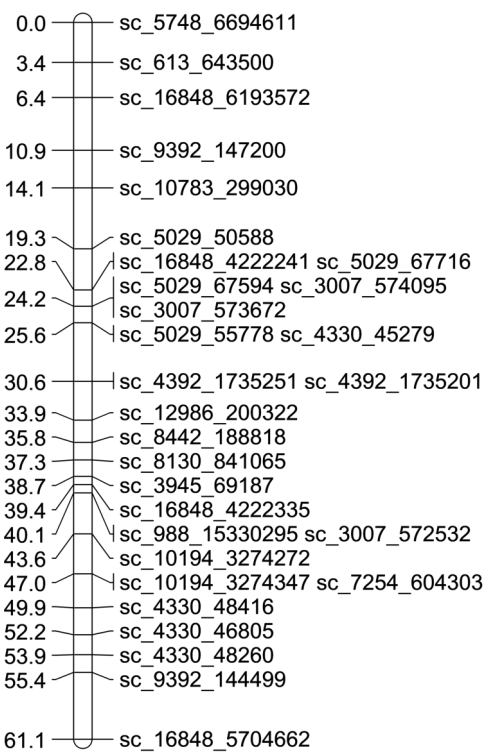
(h) 8vo grupo de ligamiento con 39 loci.



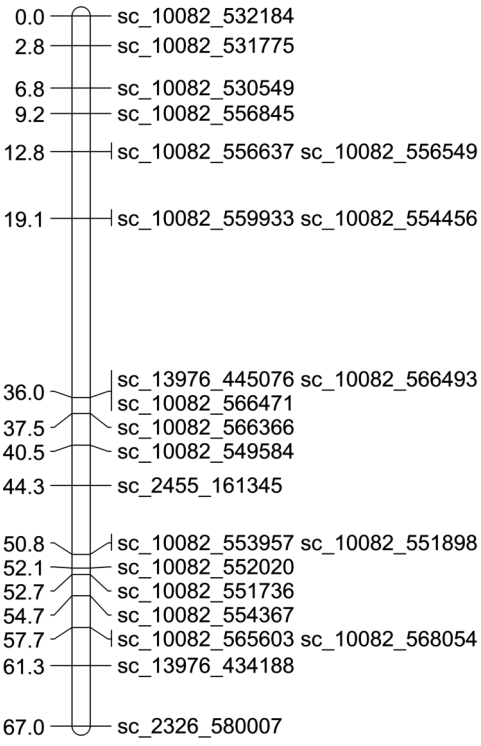
(i) 9no grupo de ligamiento con 36 loci.



(j) 10mo grupo de ligamiento con 35 loci.

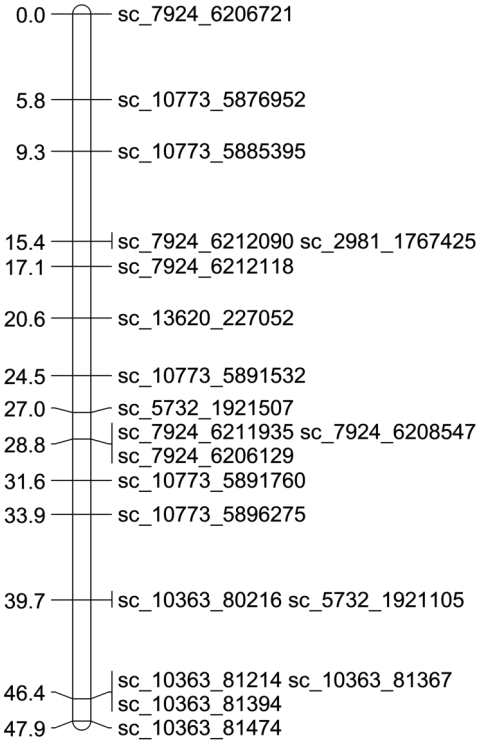


(k) 11ro grupo de ligamiento con 30 loci.

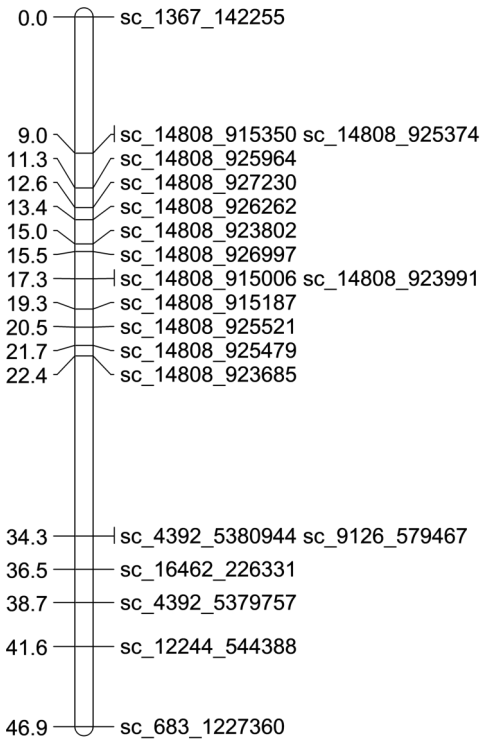


(l) 12do grupo de ligamiento con 23 loci.

Figura 3.14: grupos de ligamiento 7 a 12 del mapa genético generado con JM. Estrategia B con un indLOD de 15.0

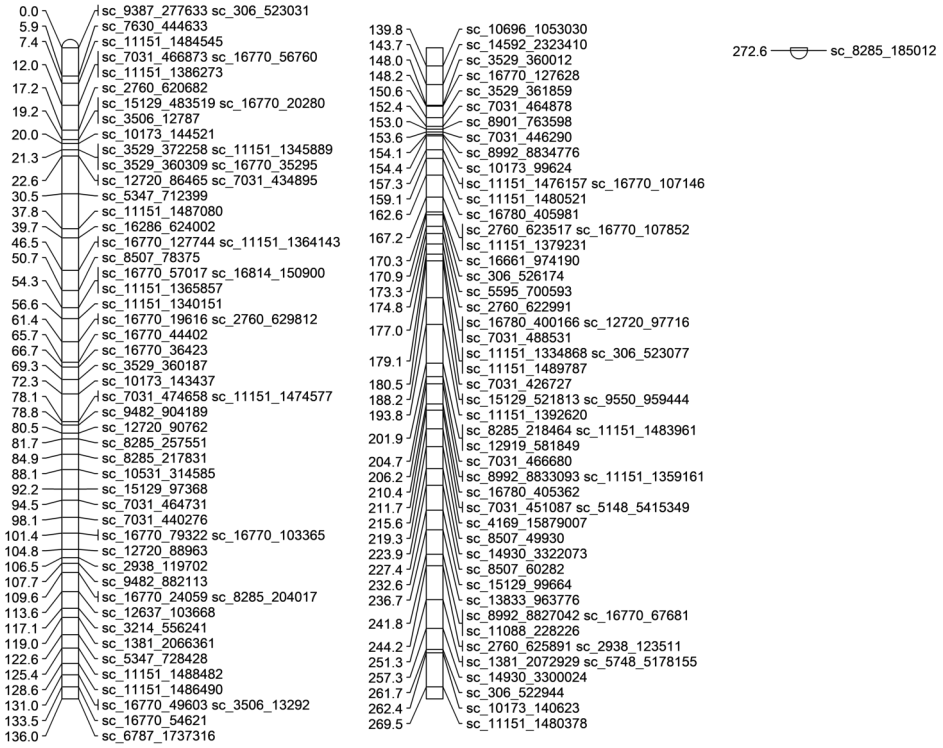


(m) 13ro grupo de ligamiento con 20 loci.

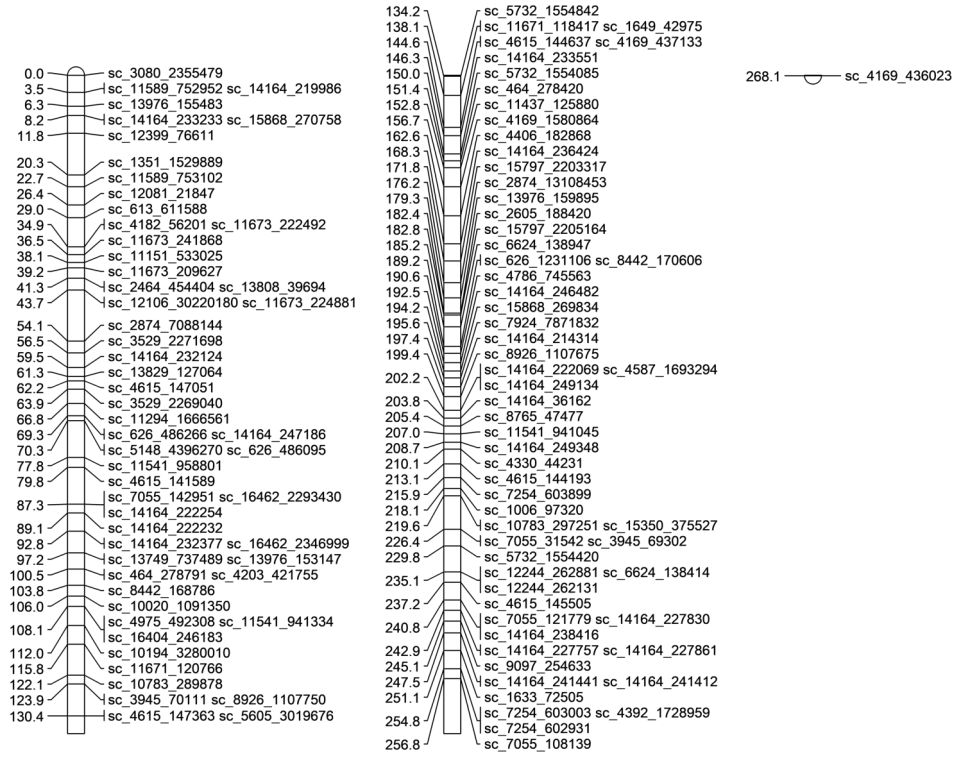


(n) 14to grupo de ligamiento con 20 loci.

Figura 3.14: grupos de ligamiento 13 y 14 del mapa genético generado con JM. Estrategia B con un indLOD de 15.0

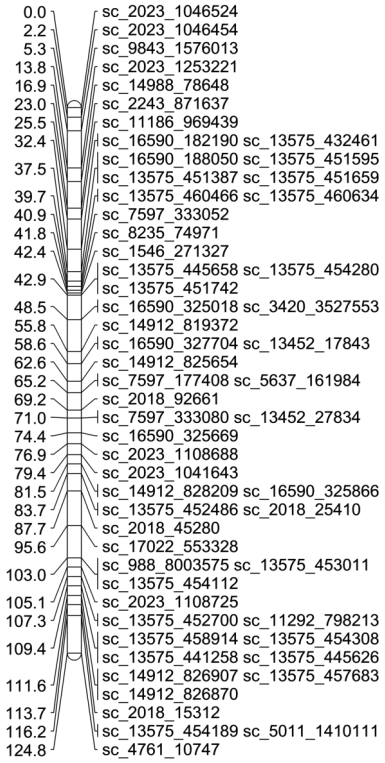


(a) 1er grupo de ligamiento con 119 loci.

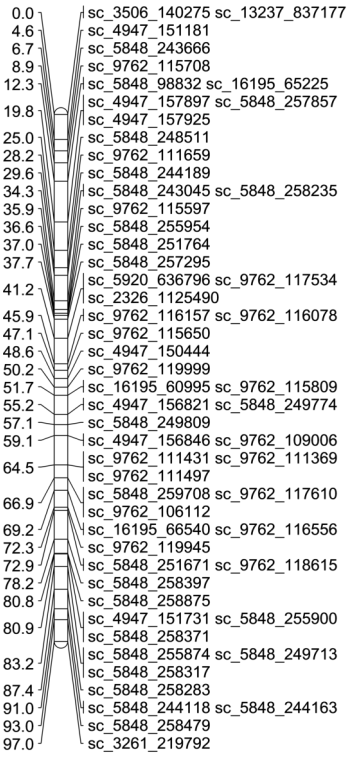


(b) 2do grupo de ligamiento con 115 loci.

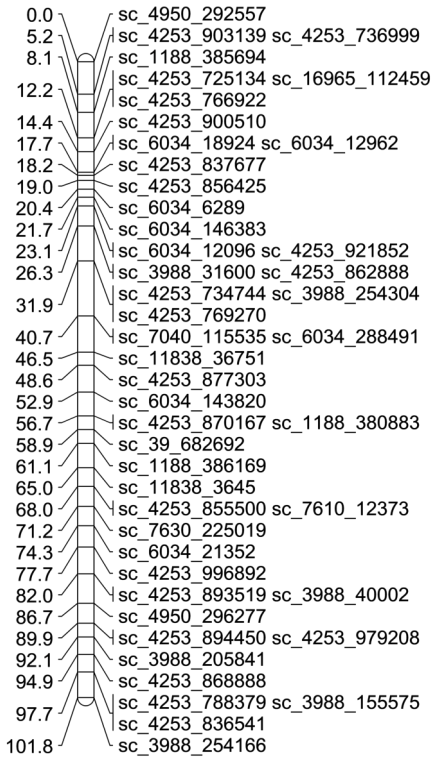
Figura 3.15: grupos de ligamiento 1 a 2 del mapa genético generado con JM. Estrategia C con una RF de 0.015



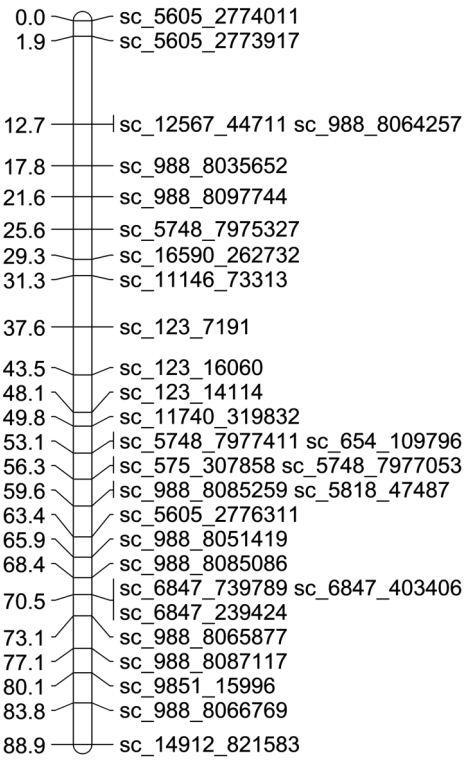
(c) 3er grupo de ligamiento con 58 loci.



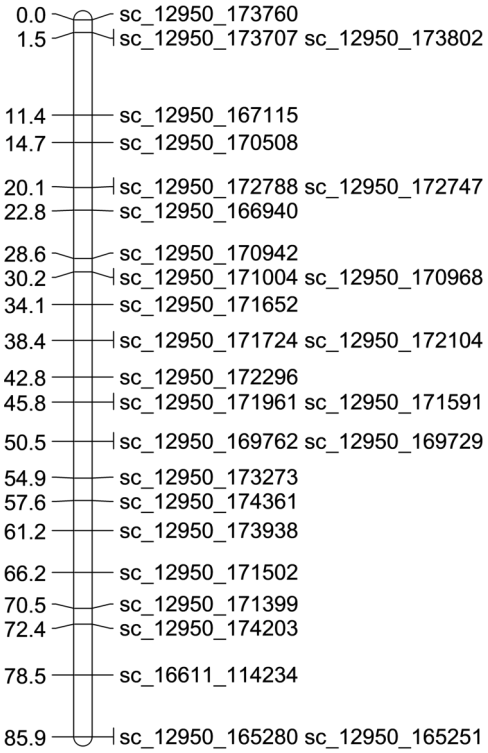
(d) 4to grupo de ligamiento con 58 loci.



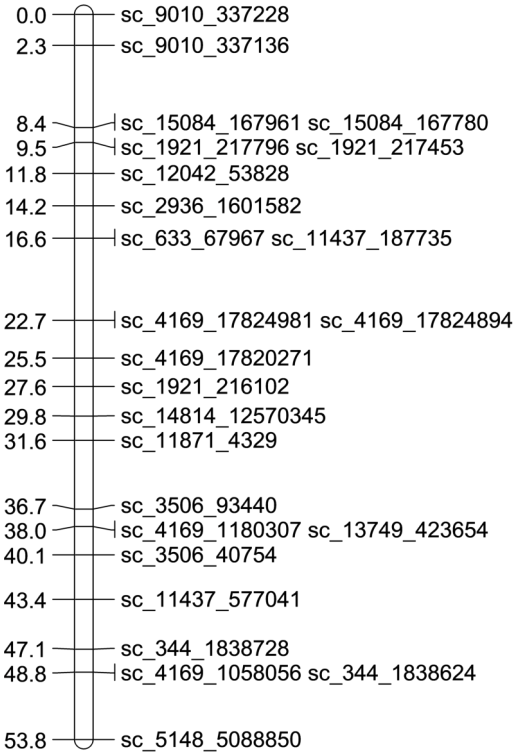
(e) 5to grupo de ligamiento con 47 loci.



(f) 6to grupo de ligamiento con 30 loci.

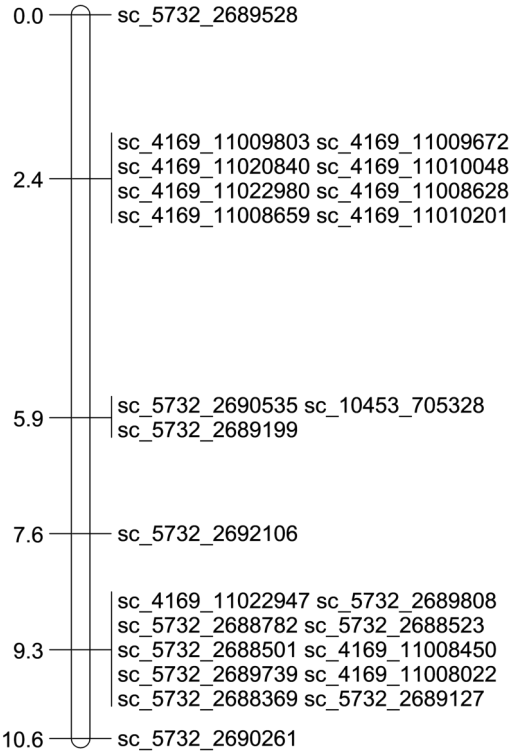


(g) 7mo grupo de ligamiento con 28 loci.

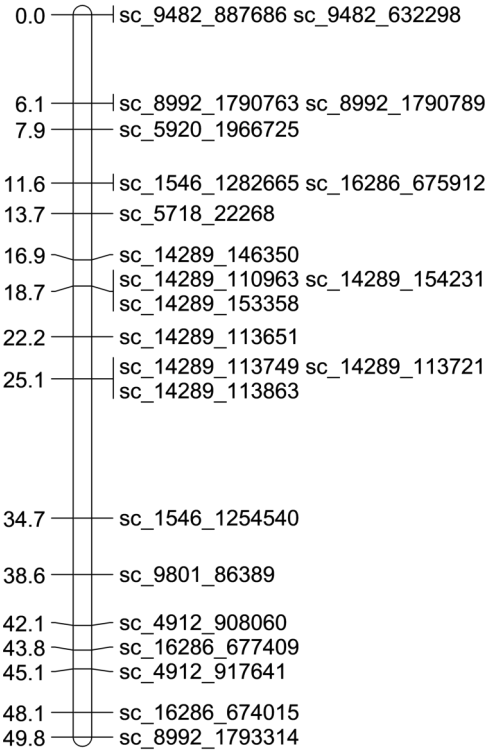


(h) 8vo grupo de ligamiento con 25 loci.

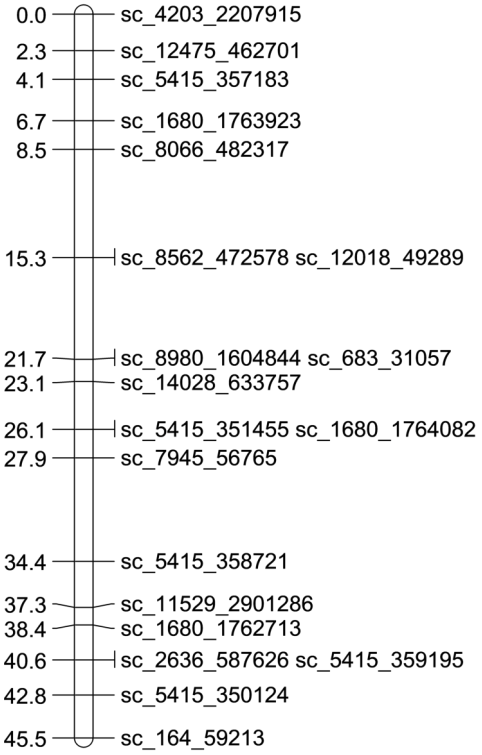
Figura 3.15: grupos de ligamiento 3 a 8 del mapa genético generado con JM. Estrategia C con una RF de 0.015



(i) 9no grupo de ligamiento con 24 loci.



(j) 10mo grupo de ligamiento con 23 loci.



(k) 11ro grupo de ligamiento con 20 loci.

Figura 3.15: grupos de ligamiento 9 a 11 del mapa genético generado con JM. Estrategia C con una RF de 0.015

Con estas estadísticas, se hicieron dos tablas comparativas de métricas sobre los mapas generados. En la tabla 3.7 se muestran estas métricas, además, del ensamblaje hecho hasta el nivel de *scaffolds*. En esa tabla hay estadísticas como el número de *scaffolds* asociados en cada estrategia (**# Scaffolds**). En la siguiente fila está el número de grupos de ligamiento (GL) o grupos de cosegregación (Garsmeur *et al.*, 2018) (**# Grupos de ligamiento**). La siguiente entrada corresponde al tamaño total del ensamblaje o mapa (**Tamaño**). El N50 aparece en la cuarta fila, el cual es una cifra que representa el tamaño del ensamblaje según la secuencia (en este caso GL) más corta, que sumada en orden descendente con el tamaño de sus predecesores GL, es al menos el 50 % del tamaño total del ensamblaje (**N50**), si el N50 es de mayor valor, indica que el ensamblaje es más completo. El n:N50 es incluido y es el número de GL que tienen el tamaño de N50 (**n:N50**). El valor mínimo, que es el GL de menor tamaño (**min**), y el valor máximo, que es el GL de mayor tamaño (**max**), aparecen en la penúltima y última filas de la tabla, respectivamente.

En la tabla 3.7 se reporta que el más grande es el construido con la estrategia de filtros A y es el que tiene un mayor número de *scaffolds* asociados, pero su primer grupo de ligamiento es inclusive más grande en número de pares de base que el resto de mapas genéticos, y en la tabla 3.8, para la misma estrategia, se observa que tiene un *scaffold* que se repite en 9 grupos de ligamiento. En esta tabla 3.7, el número 1 quiere decir que el *scaffold* aparece solo una vez en un grupo de ligamiento del mapa. El número 2 quiere decir que el *scaffold* aparece dos veces en el mapa en dos grupos de ligamiento diferente y así sucesivamente hasta un máximo de nueve que hubo de *scaffolds* repetidos.

En estas tablas 3.7 y 3.8 se puede observar también que la estrategia B tiene el menor número de *scaffolds* pero no el menor número de grupos de ligamiento que conforman el mapa genético, lo que puede indicar que los *scaffolds* están menos ligados para formar más grupos. Aunque, la estrategia C tiene un tamaño mayor que la estrategia B que se aproxima más al tamaño del genoma monoploide reportado para la variedad CC 01-1940, 1.019 Gbp (Trujillo, 2020). Esto sugiere que hay un mayor número de *scaffolds* ligados en esta estrategia. Esta estrategia C tiene, también, su N50 mayor que la estrategia B por más del doble. Sin embargo, la estrategia B generó GLs de mejor tamaño y significativos por los loci incluidos y que, probablemente, los *scaffolds* relacionados son más grandes. El GL mínimo de la estrategia B es al menos 3.5 veces más grande que el GL mínimo de la estrategia C. El GL mínimo de la estrategia C es casi 28 veces más pequeño que el *scaffold* más grande ensamblado en la Version 3 (PacBio + Hi-C) del ensamblaje, pero el GL máximo de esta estrategia es casi 3 veces más grande que el GL máximo de la estrategia B, lo cual puede explicarse porque hay menos GLs en la estrategia C y hay casi 30 *scaffolds* de diferencia con la estrategia B que pueden estar ligados.

En cuanto a porcentajes de *scaffolds* repetidos en los GLs, la estrategia A tiene la mayor representación de repetidos con respecto al total con 19.25 %. En ese orden, sigue la estrategia B con 12.76 % y luego la estrategia C con 11.24 %. Aunque, la estrategia A tiene datos atípicos al presentar que 1 *scaffold* está repetido 9 veces y 2 *scaffolds* están repetidos 5 veces en diferentes GLs.

Tabla 3.7: comparativo de métricas de ensamblaje antes de los mapas genéticos y los construidos con las estrategias A, B y C.

Métrica	Versión 3 (PacBio + Hi-C)	A	B	C
# Scaffolds	17,098	449	141	169
# Grupos de ligamiento	NA	18	14	11
Tamaño	1,253 Mbp	1,397 Mbp	434.4 Mbp	503.1 Mbp
N50	1.23 Mbp	97.86 Mbp	46.25 Mbp	93.84 Mbp
n:N50	186	2	5	2
Min	1,000 bp	12.72 Mbp	3.77 Mbp	1.071 Mbp
Max	29.81 Mbp	611.7 Mbp	59.05 Mbp	171.6 Mbp

Tabla 3.8: paralelo de los mapas genéticos para comparar cuántos scaffolds repetidos tiene cada uno en sus grupos de ligamiento.

Mapas / sc repetidos	A	B	C
1	363.0	123.0	149.0
2	61.0	15.0	17.0
3	16.0	3.0	2.0
4	6.0	0.0	1.0
5	2.0	0.0	0.0
6	0.0	0.0	0.0
7	0.0	0.0	0.0
8	0.0	0.0	0.0
9	1.0	0.0	0.0
Total	**449.0**	**141.0**	**169.0**

Discusión

En este trabajo se realizó la construcción de tres mapas genéticos, debido a la importancia que tiene en el ensamblaje del genoma monoploide de la variedad comercial CC 01-1940 de caña de azúcar, y dadas las necesidades de la plataforma de mejoramiento genético de CENICAÑA. Estos mapas genéticos se construyeron a partir de una población biparental resultado del cruce entre CC 01-746 con CC 01-1940, parentales que presentan características contrastantes. Los marcadores moleculares SNPs usados para este mapa genético derivaron del alineamiento de las secuencias de la población biparental a los 17,098 *scaffolds* creados para CC 01-1940 (Trujillo, 2020).

El conjunto de lecturas pareadas, usado en este trabajo y producido a partir de la población biparental, ofrece un ensamblaje preciso para las regiones repetitivas que son menores a la distancia interna que hay entre la lectura *forward* y *reverse* de un individuo (Sims *et al.*, 2014). Las secuencias de esta progenie fueron obtenidas a partir de datos Illumina WGS. La obtención de estos datos significó un esfuerzo y recurso considerable de CENICAÑA para construir el mapa genético, dado que esta aproximación es más costosa (Sims *et al.*, 2014) frente a las tecnologías previamente usadas como RADseq y GBS.

La tecnología RADseq usa enzimas de restricción y sonicación para dividir el genoma en fragmentos de ADN, los cuales, según su tamaño, son secuenciados en plataformas de secuenciación de próxima generación (Wickland *et al.*, 2017). La tecnología GBS usa enzimas de restricción sensibles a los sitios de metilación del ADN para seleccionar y filtrar regiones altamente repetitivas del genoma (lo que introduce un sesgo), y luego se producen lecturas cortas, por secuenciación de alto rendimiento, de las regiones cercanas al clivaje de los sitios de restricción (Darrier *et al.*, 2019). Sin embargo, estas dos tecnologías son notorias porque pueden generar datos con ruido a menor profundidad de secuencia. Los datos GBS tienden a tener una gran fracción de datos faltantes que requieren imputación y, a veces, hace falta una compleja interpretación computacional previa al análisis posterior (Darrier *et al.*, 2019). Los datos RADseq no muestrearán con precisión los bloques de haplotipos con suficientes marcadores para proporcionar una cobertura de todo el genoma debido a que los posibles SNPs detectados en un bloque son usualmente redundantes con otros SNPs ya existentes en el mismo bloque de haplotipo (Lowry *et al.*, 2017).

La tecnología WGS de alta profundidad es el *gold standard* para la resecuenciación del ADN porque puede interrogar todos los tipos de variantes: SNV, indel, variantes estructurales y *copy number variation* (CNV), tanto en el genoma que codifica para proteínas como en el restante de las secuencias no codificantes (Sims *et al.*, 2014). Esta tecnología, además, incrementa la probabilidad de detectar potenciales SNPs que están en desequilibrio de ligamiento completo ($R^2 = 1$), los cuales son importantes para identificar adaptaciones a factores locales o características de interés agronómico en las poblaciones experimentales (Lowry *et al.*, 2017). Hay diferentes investigaciones que destacan las ventajas y usos de datos WGS para el ADN genómico de los híbridos de caña de azúcar y la necesidad de tener genomas de referencia disponibles para caña (Berkman *et al.*, 2014; Garsmeur *et al.*, 2018; N. Piperidis y D'Hont, 2020; Souza *et al.*, 2011; Thirugnanasambandam *et al.*, 2018). Este conjunto de datos producido por el Centro fue el insumo fundamental para el

desarrollo de este trabajo.

Este manuscrito muestra tres estrategias comparativas y diferentes basadas en un proceso de filtros previo al análisis de ligamiento para la construcción de los tres mapas genéticos. La tres estrategias tienen sus ventajas y desventajas según los conceptos y parámetros usados. La primera y la segunda usando conceptos sobre la frecuencia del alelo menor (MAF), la heterocigosidad observada (OH) y la desviación del equilibrio Hardy-Weinberg (HWE) en una población F1. La tercera, al seguir la definición de los marcadores de dosis única (*single dose markers*, SDMs) (Balsalobre *et al.*, 2017; Bourke, Voorrips *et al.*, 2018; Garsmeur *et al.*, 2018; Wu *et al.*, 1992). Es decir, cuando uno de los parentales es heterocigoto y el otro es homocigoto, los marcadores de dosis única son aquellos que se segregan en la población en una relación 1:1 (presencia:ausencia, marcador presente en una mitad de la población y ausente en la otra mitad) que representan los marcadores heterocigotos en uno de los parentales. Sin embargo, cuando ambos parentales son heterocigotos, la segregación de los SMDs es 3:1 (marcador presente en el 75 % de la población, pero ausente en el 25 % restante) que representan los marcadores heterocigotos en ambos padres (Wu *et al.*, 1992; J. Zhang, Zhou *et al.*, 2013). Este criterio se cumple a pesar que una especie sea alopoliploide, autopoliploide o diploide (Wu *et al.*, 1992).

Esta teoría aplicada a los loci derivados de secuenciación, presentes en este trabajo, se implementa a través de la prueba chi-cuadrado al contar los genotipos (heterocigoto u homocigoto) de cada individuo en la población por cada marcador. Por ejemplo, asumiendo que se ha escogido que el locus en particular sea heterocigoto para uno de los parentales (el de interés) y homocigoto para el otro, si el valor P de este locus supera un umbral establecido, quiere decir que el marcador no se desvía significativamente de los valores esperados. Es decir, no hay evidencia significativa para rechazar la hipótesis de que la mitad de los individuos son homocigotos y la otra mitad son heterocigotos cuando un parental es heterocigoto y el otro no. Esto sugiere entonces que se escoge el marcador porque es un SDM con proporción 1:1 útil para el mapa genético del parental de interés.

En la tercera estrategia implementada, dada la población biparental, se seleccionaron los loci que para CC 01-1940 eran heterocigotos, pero homocigotos para CC 01-746, entonces al menos la mitad (50 %) (o un valor cercano) de los individuos deberían ser heterocigotos, y la otra mitad (o un valor cercano) deberían ser homocigotos para que el loci fuera seleccionado. Estos serían los loci heterocigotos de CC 01-1940 para construir un mapa genético de esta variedad. Los SDMs son abundantes en genomas autoploides, representan el 70 % de los loci polimórficos (Silva *et al.*, 1993). El concepto de los SDMs fue usado en los trabajos de Garsmeur *et al.* (2018) y Balsalobre *et al.* (2017) y en al menos 10 mapas genéticos creados para caña de azúcar usando diferentes tecnologías de marcadores moleculares (J. Zhang, Zhou *et al.*, 2013).

La ventaja de los SDMs es que se heredan similarmente a través de cualquier nivel de ploidía y comportamiento de recombinación, lo que permite el uso de un *software* de mapeo diploide (Bourke, Voorrips *et al.*, 2018). Lo anterior es un beneficio adicional para una población F1, la cual se establece rápidamente (solo se requiere el cruzamiento de la primera generación) y se pueden crear dos mapas genéticos, uno para cada parental. En el primer mapa se escogen los loci heterocigotos en un parental, pero homocigoto en el otro (mapa genético del parental 1), y viceversa (mapa genético del parental 2). Esto es válido debido a que no hay meiosis entre ambos padres (J. Zhang, Zhou *et al.*, 2013). La aplicación de los SDMs en una F1 no es la única aproximación para hacer la construcción de un mapa genético para poliploides con secuenciación de alto rendimiento. En este trabajo se mostró que usando el MAF, la OH y el HWE, también se puede construir un mapa genético potencial que represente el genoma monoploide de la variedad CC 01-1940.

La ley Hardy-Weinberg establece que dados los siguientes supuestos: una población grande con emparejamiento aleatorio, que no es afectada por las mutaciones, migraciones o selección natural, (1) la reproducción de los parentales no es un factor que altera las frecuencias alélicas o genotípicas,

y (2) las frecuencias alélicas determinan las frecuencias de los genotipos (Pierce, 2016). Del primer enunciado se puede agregar que se requiere la primera generación de emparejamiento aleatorio (en este caso, los parentales) para producir las propociones Hardy-Weinberg de los genotipos, después, tanto las frecuencias genotípicas como las alélicas no cambian con el tiempo si la población sigue cumpliendo los supuestos de la ley.

Si una progenie cumple estas suposiciones para un locus, se puede decir que ese locus está en HWE. En la población, un locus puede estar en HWE, pero no necesariamente los demás. Una implicación para un locus bialélico en HWE es que su frecuencia de heterocigotos es mayor cuando las frecuencias alélicas (frecuencia del alelo menor versus la frecuencia del alelo mayor) están entre 0.33 y 0.66, y alcanza su punto máximo, no será mayor si mantiene el HWE, cuando las frecuencias alélicas son 0.5 (Pierce, 2016). Bajo estas premisas, en este trabajo se seleccionaron aquellos marcadores que no tuvieran una desviación significativa del HWE, es decir, después de ejecutarse una prueba chi-cuadrado, se tomaron aquellos cuyo valor P fuera mayor de 0.01. Además, se muestran los resultados de ambos métodos, siguiendo la definición de SDMs y siguiendo la noción del HWE. Ambas aproximaciones buscaron que los SNPs seleccionados pudieran ser usados en un *software* para construir mapas genéticos en organismos diploides.

En esta investigación se evaluaron varios *softwares* presentados en el trabajo de Bourke, Voorrips *et al.* (2018) y se establecieron parámetros relacionados con la antigüedad y uso experimental en poblaciones de caña de azúcar. JoinMap v5.0 (JM) (Van Ooijen, 2018) fue el *software* escogido por tener, principalmente, un gran número de citas y trabajos relacionados con caña de azúcar (Garcia *et al.*, 2006; Gazaffi *et al.*, 2014; Hoy *et al.*, 2016; Kilian *et al.*, 2005; Oliveira, 2006). En particular, JM ha sido usado en una investigación referente para este trabajo, Garsmeur *et al.* (2018), aunque en Balsalobre *et al.* (2017) usan OneMap como *software*, también usan la frecuencia de recombinación y el valor LOD. Ambos estudios construyeron un mapa genético para híbridos de caña. A pesar de esto, una posibilidad para explorar otros resultados podría ser el uso de los *softwares* polymapR y OneMap, ambos con una continua liberación de versiones, que podrían estimar unas distancias genéticas más precisas o que podrían incluir un mejor orden de los marcadores dado el desequilibrio de ligamiento. Para este trabajo, el número de referencias sustentado para JM validó hacer el análisis de ligamiento de los loci con este *software*.

Los marcadores en JM fueron usados para la construcción de los mapas genéticos evaluando diferentes combinaciones de *Logarithm of Odds* (LOD) independiente. El LOD es el logaritmo en base 10 de la relación entre la probabilidad de tener la segregación de un par de loci suponiendo que están ligados y la probabilidad de obtener esa segregación pero ahora suponiendo que el par de loci no están ligados. El valor LOD se convierte en un estimado estadístico para saber si dos marcadores (SNPs) tienen alta probabilidad de estar cerca uno del otro en el mismo cromosoma y, por tanto, ser heredados juntos. Por regla general, un valor LOD de 3 indica que dos marcadores están cerca uno del otro en el mismo cromosoma, es decir, entre más alto sea el valor LOD, hay más certeza de que dos marcadores estén realmente ligados y sean heredados juntos. En particular, un valor LOD de 3 quiere decir que la probabilidad de ligamiento, que ocurre a cierta distancia en centiMorgan, es 1,000 veces mayor que la probabilidad de no ligamiento.

El LOD es calculado a partir de la frecuencia de recombinación (FR), la cual consiste en, dado un par de SNPs, la probabilidad de que un cruzamiento de los cromosomas ocurra entre este par de SNPs y se calcula como el cociente entre la cantidad de individuos en la progenie que presentan un rasgo resultado de la recombinación de dos alelos (marcadores) y el total de individuos. El LOD independiente (indLOD) es un parámetro que puede manipularse desde la interfaz de JM y no es afectado por la distorsión de la segregación como la estimación del LOD, empleada a menudo en el análisis de ligamiento, lo que conduce a una menor incidencia de vinculación errónea (Van Ooijen, 2018).

Los tres mapas genéticos se construyeron usando JM con diferentes valores de indLOD, la FR y usando el algoritmo de mapeo de máxima verosimilitud, el cual consiste en un proceso de construcción del mapa al adherir loci usando muestreo espacial (*spatial sampling*) y, posteriormente, encontrar el mejor orden para el conjunto de loci obtenidos (Van Ooijen, 2018). Al implementar cada una de las tres estrategias o mapas se obtuvieron 3,739 loci, a 4,096 loci y a 7,816 loci, respectivamente. Estos diferentes conjuntos de marcadores fueron suficientes para construir los mapas genéticos. En otros estudios, por ejemplo, en Balsalobre *et al.* (2017) construyeron un mapa genético con 934 marcadores SNPs provenientes de una progenie de 151 hermanos completos resultado del cruce de dos variedades SP80-3280 (madre) y RB835486 (padre). Garsmeur *et al.* (2018) construyeron un mapa genético con 13,062 SNPs provenientes de una población de 186 individuos autoprogenie de la variedad R570 y J. Zhang, X. Zhang *et al.* (2018) construyeron un mapa genético con 998,370 SNPs provenientes de una población recruzada de 54 individuos resultado del cruce entre el doble haploide AP83-108 y su progenitor octoploide SES208.

En las diferentes estrategias propuestas que arrojan los mapas genéticos candidatos para representar el genoma monoploide de la variedad CC 01-1940, no se encontraron ventajas que pudieran seleccionar a una sola. Cada mapa genético tiene una métrica que lo destaca, ya sea: el tamaño total del mapa en pares de base o en centiMorgans, el N50, la cantidad de grupos de ligamiento, el tamaño de los grupos de ligamiento (kbp o cM), el número de *scaffolds* repetidos, la cantidad de loci mapeados, la saturación del mapa por densidad cM/loci o kbp/loci o la relación del cociente entre densidades en kbp o cM. Por ejemplo, en J. Zhang, X. Zhang *et al.* (2018) construyeron un mapa genético de 451.3 Mbp, en Balsalobre *et al.* (2017) generaron un mapa de 1,424.94 cM con microsatélites o SSR (*Simple Sequence Repeats*) y marcadores GBS, y en Garsmeur *et al.* (2018) lograron un ensamblaje de las regiones ricas en genes de 382 Mbp. Estos valores mencionados son poco distantes a los reportados en este trabajo.

Para los tres mapas genéticos se debe establecer un criterio de selección que determine cuál puede ensamblar con mejor continuidad el conjunto de los 17,098 *scaffolds*. Para este propósito se puede hacer un análisis de colinealidad contra la anotación de genes de los genomas de *S. spontaneum* (J. Zhang, X. Zhang *et al.*, 2018) y el híbrido R570 (Garsmeur *et al.*, 2018). Una hipótesis inicial es que, aunque para ambos análisis se debe esperar sintenia, el experimento contra las regiones ricas en genes del ensamblaje de R570 podría tener un mejor comportamiento porque es un híbrido al igual que CC 01-1940. Los híbridos de caña comparten un 80 % de material genético de *S. officinarum*, pero tan solo 10-15 % de *S. spontaneum* (Cuadrado *et al.*, 2004; D'Hont, 2005; D'Hont, Grivet *et al.*, 1996; G. Piperidis *et al.*, 2010).

Finalmente, los resultados de este trabajo son las primeras versiones de mapas genéticos creados en CENICAÑA como resultado de un proceso de secuenciación de alto rendimiento. Por tanto, es una metodología novedosa para el Centro que puede ser usada en futuras ocasiones. Se podría intentar con métricas favorables para producir mapas genéticos que tengan mayor continuidad, luego de ver la colinealidad con otros genomas de referencia. Por ejemplo, se podrían mover los rangos del MAF y la OH para incluir más loci en el mapa de *scaffolds* importantes para el ensamblaje. Si se usa JM, podría variarse el valor indLOD o la FR para incrementar el número de marcadores presentes en el mapa. La estrategia B y la C fueron exigentes en indLOD y FR, respectivamente, y sus estadísticas son más similares en comparación con la estrategia A. Se debe notar que después de la fase de filtros, la estrategia C arrojó más loci que las demás. Una oportunidad sería combinar los filtros de estas últimas estrategias y evaluar los resultados, por ejemplo, SNPs que estén en HWE, pero que también, en el conteo de heterocigotos y homocigotos, no se desvíen significativamente del 50 % esperado. Por último, se podría explorar la construcción del mapa con alguno de los *softwares* evaluados en este trabajo y ver que el análisis de vinculación por desequilibrio de ligamiento se comporte mejor que lo visto en JM.

Información suplementaria

A.1. Lista de variedades producidas por CENICAÑA

Tabla A.1: Listado de variedades producidas por o en colaboración con CENICAÑA. CC: Cenicaña Colombia. SP: São Paulo. [1]

Variedades de Cenicaña Colombia		
CC 01-746	CC 87-434	CC 97-7170
CC 00-3257	CC 86-33	CC 94-5827
CC 11-600	CC 85-96	CC 93-7510
CC 05-430	CC 85-92	CC 93-4418
CC 11-595	CC 85-68	CC 93-4223
CC 10-450	CC 85-63	CC 93-4181
CC 11-605	CC 84-75	CC 93-3895
CC 04-195	CC 84-66	CC 93-3826
CC 09-874	CC 84-56	CC 93-3803
CC 09-702	CC 84-10	CC 93-744
CC 99-2282	CC 83-25	CC 92-2804
CC 02-3257	CC 82-28	CC 92-2393
CC 09-535	CC 82-27	CC 92-2358
CC 09-066	CC 82-26	CC 92-2198
CC 06-783	CC 82-15	CC 92-2188
CC 03-469	CC 03-154	CC 92-2154
CC 05-948	CC 01-1940	CC 91-1945
CC 05-230	CC 01-1228	CC 91-1880
CCSP 89-259	CC 01-678	CC 91-1606
CCSP 89-43	CC 00-3771	CC 91-1555
CC 89-2000	CC 00-3079	CCSP 89-1997
CC 87-505	CC 98-72	

A.2. Evaluación y alineamiento de las muestras

A.2.1. Usar el software FastQC para evaluar las lecturas

Algoritmo A.1: instrucción usada para hacer la evaluación de las secuencias enviadas por NOVOGENE. [5]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 01_eval_samples.sh
# Fecha: 18-09-2020
# Objetivo: Evaluar la calidad de las secuencias con el software FastQC v11.8.
# Salida: Resultado de la ejecución de los módulos que analiza el software para cada uno del conjunto de lecturas forward y reverse de un individuo. Dos archivos .html y dos archivos .zip por cada lectura .
# Entrada: Ingresar ambos conjuntos de lecturas forward y reverse .fq.gz.


cat ../../data/sample_ids.txt | \ #El archivo sample_ids.txt tiene los nombres de las muestras de lectura de los individuos, al ejecutar 'cat' sobre este, su salida (usando | (pipe)) es usada en la instrucción parallel para ejecutar fastqc. Ejemplo del contenido de este archivo...
:'
            P2_277_USD16089481L_HJJNWDSXX_L4
            P2_300_USD16089488L_HJNMNDSXX_L3
            P2_60_USD16089417L_HJM27DSXX_L1
            P2_7_USD16089400L_HJNMNDSXX_L3_L4
            P2_74_USD16089423L_HJM27DSXX_L2
            P2_80_USD16089425L_HJM27DSXX_L2
            CC_011940_USD16089495L_HJJNWDSXX_L4
'
parallel --gnu -j50 'fastqc ../../data/{}/*.fq.gz -o {}/'
# La opción --gnu se usa por compatibilidad para comportarse como GNU parallel; # -j50 es para correr 50 tareas en paralelo (con esto se logra acelerar la ejecución del comando);
# ../../data/ ruta relativa donde están guardados los archivos de la lecturas;
# {} reemplaza la salida del archivo sample_ids.txt línea por línea en la instrucción fastqc para ser ejecutada;
# *.fq.gz todos los archivos cuyos nombres terminen en .fq.gz;
# -o de fastqc significa el nombramiento al archivo de salida.
```

A.2.2. Valores Phred por individuo y lectura

Tabla A.2: descripción de los valores Phred que en promedio se obtuvieron por cada individuo en cada par de sus conjuntos de lectura *forward* o *reverse*. En azul está resaltado el individuo-lectura con el valor máximo y en rojo aquel con el valor mínimo. [13]

P2-id-lectura	Valor								
P2_300_2	36.1315	P2_76_2	35.8114	P2_89_1	36.0504	P2_43_1	36.0645	P2_311_2	36.0071
P2_300_1	36.3957	P2_76_1	36.0362	P2_89_2	35.9248	P2_43_2	35.8934	P2_311_1	36.2055
P2_80_1	36.0657	P2_227_1	36.0659	P2_216_1	36.0903	P2_69_2	35.763	P2_310_2	35.7224
P2_80_2	35.7337	P2_227_2	35.7636	P2_216_2	35.679	P2_69_1	36.0693	P2_310_1	36.2173
P2_7_2	36.2085	P2_245_1	36.2785	P2_57_1	36.0561	P2_81_2	35.8208	P2_199_1	36.3119
P2_7_1	36.4131	P2_245_2	36.0485	P2_57_2	35.8918	P2_81_1	36.0837	P2_199_2	36.133
P2_277_2	35.889	P2_232_1	36.1059	P2_103_1	36.0552	P2_28_2	35.6997		
P2_277_1	36.208	P2_232_2	35.9422	P2_103_2	35.8019	P2_28_1	36.0278		
P2_66_1	36.0539	P2_166_1	36.0682	P2_305_1	36.188	P2_88_2	35.7247		
P2_66_2	35.8044	P2_166_2	35.8386	P2_305_2	36.0008	P2_88_1	36.03		
P2_74_1	36.0694	P2_210_2	35.7408	P2_201_1	36.311	P2_129_2	35.6199		
P2_74_2	35.8049	P2_210_1	36.0534	P2_201_2	36.1725	P2_129_1	36.0883		
CC_011940_2	35.9873	P2_44_1	36.0586	P2_65_1	36.0258	P2_52_1	36.0557		
CC_011940_1	36.2179	P2_44_2	35.8443	P2_65_2	35.5946	P2_52_2	35.6586		
P2_188_2	35.5589	P2_278_1	36.2259	P2_111_2	35.8799	P2_164_1	36.0912		
P2_188_1	36.0828	P2_278_2	36.0088	P2_111_1	36.0867	P2_164_2	35.7567		
P2_168_1	36.0766	P2_184_1	36.0816	P2_263_2	35.8227	P2_259_2	35.9969		
P2_168_2	35.9122	P2_184_2	35.792	P2_263_1	36.2427	P2_259_1	36.2716		
P2_271_1	36.2045	P2_55_2	34.5809	P2_295_2	35.945	P2_237_2	35.9763		
P2_271_2	35.9425	P2_55_1	36.0484	P2_295_1	36.2259	P2_237_1	36.3025		
P2_262_1	36.2765	P2_207_1	36.0708	P2_206_2	35.7477	P2_290_2	36.2163		
P2_262_2	36.0454	P2_207_2	35.6542	P2_206_1	36.0903	P2_290_1	36.3836		
P2_142_2	35.609	P2_84_2	35.5607	P2_49_1	36.0178	P2_225_1	36.0797		
P2_142_1	35.9089	P2_84_1	36.0597	P2_49_2	35.7705	P2_225_2	35.59		
P2_144_2	35.7493	P2_256_2	36.037	P2_180_1	36.0666	P2_296_2	35.8157		
P2_144_1	36.0803	P2_256_1	36.2911	P2_180_2	35.8167	P2_296_1	36.1965		
P2_270_2	36.0143	P2_147_2	35.8235	P2_42_2	35.9069	P2_153_1	36.0826		
P2_270_1	36.3026	P2_147_1	36.0698	P2_42_1	36.0542	P2_153_2	35.6314		
P2_60_1	36.0239	P2_114_2	35.9588	P2_231_2	35.7703	P2_167_2	35.7917		
P2_60_2	35.7824	P2_114_1	36.1024	P2_231_1	36.0834	P2_167_1	36.0816		
P2_171_1	36.0995	P2_87_2	35.5307	P2_250_1	36.2922	P2_187_2	35.7839		
P2_171_2	35.9252	P2_87_1	36.0582	P2_250_2	36.0948	P2_187_1	36.076		
P2_145_2	35.7637	P2_154_2	36.2033	P2_220_1	36.3115	P2_40_1	36.0506		
P2_145_1	36.0724	P2_154_1	36.2917	P2_220_2	36.1918	P2_40_2	35.6849		
P2_172_1	36.1066	P2_221_1	36.0705	P2_282_1	36.1992	P2_32_1	36.0144		
P2_172_2	35.7674	P2_221_2	35.9327	P2_282_2	35.7865	P2_32_2	35.8698		
P2_63_2	35.6112	P2_140_2	35.8522	P2_54_1	36.0497	CC_01746_2	36.0234		
P2_63_1	36.0258	P2_140_1	36.0927	P2_54_2	35.7857	CC_01746_1	36.2169		
P2_17_1	36.4046	P2_217_2	35.9694	P2_67_2	35.8655	P2_108_2	35.8032		
P2_17_2	36.0885	P2_217_1	36.3048	P2_67_1	36.0536	P2_108_1	36.0583		
P2_283_2	35.9529	P2_238_1	36.2822	P2_107_2	35.8639	P2_20_2	36.0633		
P2_283_1	36.2014	P2_238_2	35.6203	P2_107_1	36.0756	P2_20_1	36.3991		
P2_90_2	35.6803	P2_115_2	35.7514	P2_251_1	36.2987	P2_41_2	35.8909		
P2_90_1	36.0633	P2_115_1	36.0901	P2_251_2	36.0769	P2_41_1	36.0497		
P2_11_1	36.3481	P2_307_1	36.38	P2_158_2	35.9286	P2_315_1	36.223		
P2_11_2	35.8094	P2_307_2	36.1341	P2_158_1	36.0906	P2_315_2	36.0043		

A.2.3. Cálculo del inserto

Algoritmo A.2: comandos usados para hacer el cálculo del inserto que será luego usado en los alineamientos. [5]

```
# Autor(es): Jorge Duitama y Equipo de Bioinformática de CENICAÑA.
# Editor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 02_insert_length.sh
# Fecha: 18-09-2020
# Objetivo: Calcular el tamaño del inserto para las lecturas Illumina paired-end.
# Salida: archivo .log  con dos columnas: la primera tamaño del inserto y la segunda con el número de fragmentos que hay con ese tamaño.
# Entrada: Archivos de entrada y el parámetro p es el id extenso de un individuo.

p=$1; # parámetro de entrada el cual es el id de un individuo.
o=$2; # parámetro opcional.
m=`echo ${p} | sed 's/_USD.*//g'`; # 'echo' imprime el valor de la variable $p (id de un individuo) y esta salida es la entrada para 'sed', el cual sustituye todos los caracteres que empiezan por _USD por un espacio vacío. Esto se hace para dejar solo el id del individuo de los archivos de las lecturas, por ejemplo, sustituir P2_142_USDABC123.fq.gz por P2_142. Este resultado se asigna a la variable 'm'.

# Archivos de entrada: los dos conjuntos de lectura WGS fastq del individuo
# y la referencia a la cual se alinea. El primer conjunto es forward y el segundo es reverse.
f1=../data/${m}/${p}_1.fq.gz;
f2=../data/${m}/${p}_2.fq.gz;
REFERENCE=../reference/scaffolds_final_NGSEP_polished;

 # Asignación de variables para las rutas donde están ubicados los softwares que se van a usar, por ejemplo, Bowtie2 y Samtools.
BOWTIE2=../bowtie2/bowtie2;
SAMTOOLS=../samtools-1.9/samtools;

# Extracción de las primeras 250000 lecturas del conjunto de secuencias forward.
zcat ${f1} | head -n 1000000 > ${p}_testInsert_1.fastq;
# Extracción de las primeras 250000 lecturas del conjunto de secuencias reverse.
zcat ${f2} | head -n 1000000 > ${p}_testInsert_2.fastq;
# zcat lee los archivos asignados en f1 y en f2;
# head toma el primer millón de líneas de cada archivo;
# el 1000000 se usa porque el formato FASTQ ocupa 4 líneas en el archivo por cada secuencia.
# cada uno de los resultados de head es guardado en los archivos .fastq

${BOWTIE2} --rg-id ${p} --rg SM:${p} -X 1000 ${o} -k 3 -t -x ${REFERENCE} -1 ${p}_testInsert_1.fastq -2 ${p}_testInsert_2.fastq -S ${p}_testInsert.sam >& ${p}_testInsert.log;
# bowtie2 es el software para hacer los alineamientos.
# --rg-id este parámetro interpreta al valor de la variable p para que sea registrada en los archivos de salida .fastq. Ejemplo, ID:P2_142
# --rg similar al anterior pero esta vez es con el tag SM. Ejemplo, SM:P2_142
# -X es el tamaño máximo del inserto, el cual es la suma de las dos lecturas pareadas más la distancia interna entre cada una de ellas, el 'gap' que las separa. Se sabe que las lecturas pareadas de este trabajo son de 150 bp cada una pero no hay certeza de cuánto es el 'gap' que hay entre ellas entonces se exige a Bowtie2 que busque alineamiento congruentes dentro de un tope máximo de 1000 bp. El máximo hubiera podido ser mayor, pero haría más lento el proceso y no sería necesario. El 'gap' entre ambas lecturas no será mayor que 1000 bp.
# -k indica hasta N alineaciones distintas y válidas para cada lectura, donde N es igual al entero especificado con este parámetro, por ejemplo, 3.
# -t número de caracteres usados en el algoritmo Burrows-Wheeler para comprimir el tamaño de la lectura al juntar caracteres repetidos. En este caso está el valor por defecto, 10.
```

```
# -x este parámetro sirve para fijar un nombre base al genoma de referencia, el cual está en la variable REFERENCE.
# -1 parámetro que indica el archivo del primer conjunto de lecturas (forward).
# -2 parámetro que indica el archivo del segundo conjunto de lecturas (reverse).
# -S parámetro para indicar en qué archivo se va a escribir el resultado de los alineamiento, en este caso es el valor de la variable $p con el sujifo '_testInsert.sam'
# >& instrucción para indicar que habrá un archivo .log en el que se describirá qué se está ejecutando y cómo los argumentos (anteriores) están siendo evaluados.

# Calcular la distribución del tamaño del inserto según los insertos encontrados con un tamaño máximo de 1000.
${SAMTOOLS} view -SF 268 ${p}_testInsert.sam | awk '
{
    l=$9;
    if(l>=0){
        i=sprintf("%d",l/25)+1;
        if(i<100)
            a[i]++;
        else
            aM++
    }
}
END{
    for(i=1;i<100;i++)
        print (i-1)*25,a[i];
    print "More",aM
}' >> ${p}_testInsert.log;
# samtools es el software para manipular archivos .sam
# view es la instrucción para visualizar este tipo de archivos.
# -S opcional, usado antes en versiones anteriores de Samtools cuando el archivo estaba en formato .sam y no en otros como .bam o .cram.
# -F No imprime alineamientos con más de 268 bits en el campo FLAG.
# awk recibe la salida del SAMTOOLS view para calcular una distribución del número de fragmentos según sus tamaños, y finalmente, guardarlos en el archivo ${p}_testInsert.log

 # se sugiere eliminar los archivos innecesarios; el parámetro -f forza la eliminación en caso de errores, por ejemplo, archivos no existentes.
rm -f ${p}_testInsert_1.fastq;
rm -f ${p}_testInsert_2.fastq;
rm -f ${p}_testInsert.sam;
```

A.2.4. Alineamientos

Algoritmo A.3: programas e instrucciones usadas para hacer los alineamientos. [5]

```
# Autor(es): Jorge Duitama y Equipo de Bioinformática de CENICAÑA.
# Editor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 03_run_alignment.sh
# Fecha: 18-09-2020
# Objetivo: Hacer los alineamientos de las lecturas al genoma de referencia y ordenar estos alineamientos.
# Salida: Archivos .bam y _sorted.bam por cada uno de los 95 individuos.
# Entrada: Conjunto de lecturas forward y reverse y el parámetro, el genoma de referencia y la índices calculados previamente para este genoma.

p=$1; #parámetro de entrada el cual es el id de un individuo.
i=$2; #parámetro para fijar el tamaño mínimo del inserto según lo resultado de la
```

```
    ejecución del algoritmo 02_insert_length.sh
x=$3; #parámetro para fijar el tamaño máximo del inserto según lo resultado de la
    ejecución del algoritmo 02_insert_length.sh
o=$4; #parámetro opcional.
s=SM:${p}; #variable que es igual al parámetro $p pero con una cadena adicional '
    SM:'.
sample=`echo ${p} | sed 's/_USD.*//g'`; #'echo' imprime el valor de la variable $p
     y esta salida es la entrada para 'sed', el cual sustituye todos los caracteres
     que empiezan por _USD por un espacio vacío. Esto se hace para dejar solo el id
     del individuo de los archivos de las lecturas, por ejemplo, sustituir
    P2_142_USDABC123.fq.gz por P2_142. Este resultado se asigna a la variable '
    sample'.
save_location=../results/alignment/${sample}/${p}; # la variable 'save_location'
    es asignada con la ruta absoluta en la cual se van a guardar los resultados de
    los alineamientos de cada individuo.

# Los dos conjuntos de lectura WGS fastq del individuo
# y la referencia a la cual se alinea. El primer conjunto es forward y el segundo
    es reverse.
f1=../data/${sample}/${p}_1.fq.gz;
f2=../data/${sample}/${p}_2.fq.gz;
REFERENCE=../reference/scaffolds_final_NGSEP_polished.fa; #Genoma de referencia CC
    -01-1940.

INDEXES=../reference/scaffolds_final_NGSEP_polished; # Archivo Bowtie en el que se
     guardaron los índices del genoma de referencia después de una primera alineaci
    ón con lecturas radseq. Estas alineaciones están indexadas para hacer consultas
     más rápidas.

# Asignación de variables para las rutas donde están ubicados los softwares que se
     van a usar, por ejemplo, Bowtie2 y Samtools.
BOWTIE2=../bowtie2-2.3.5/bowtie2; #ruta donde está el archivo binario de Bowtie2.
PICARD=../picard-tools-2.20/picard.jar; #ruta donde está el archivo binario de
    Picard.
SAMTOOLS=../samtools-1.9/samtools; #ruta donde está el archivo binario de Samtools
    .

JAVA="java -d64 -XX:MaxHeapSize=16g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -XX:MaxHeapSize = 16g (igual a -Xmx16g) indica la cantidad de memoria destinada
    para la ejecución de los procesos, en este caso 16 GB.

# map the reads and sort the alignment

${BOWTIE2} --rg-id ${save_location} --rg ${s} --rg PL:ILLUMINA -I ${i} -X ${x} ${o
    } -p 110 -k 3 -t -x $INDEXES -1 ${f1} -2 ${f2} 2> ${save_location}_bowtie2.log
    | ${SAMTOOLS} view -bhS - > ${save_location}_bowtie2.bam;
# bowtie2 es el software para hacer los alineamientos.
# --rg-id este parámetro interpreta al valor de la variable p para que sea
    registrada en los archivos de salida .fastq. Ejemplo, ID:P2_142
# $save_location variable cuyo valor es usado guardar el resultado de los
    alineamientos, un archivo .sam
# --rg similar al anterior pero esta vez es con el tag SM. Ejemplo, SM:P2_142
# --rg es igual al anterior pero esta vez es con el tag 'PL:ILLUMINA'
# -I es el tamaño mínimo del inserto, el cual ya fue calculado con la interpretaci
    ón de los resultados generados por 02_insert_length.sh
# -X es el tamaño máximo del inserto, el cual es la suma de las dos lecturas
    pareadas más la distancia interna entre cada una de ellas, el 'gap' que las
```

```
    separa. Este valor también se deriva de los resultados de 02_insert_length.sh
# -p indica la cantidad de procesadores que se usarán para ejecutar la instrucción
    .
# -k indica hasta N alineaciones distintas y válidas para cada lectura, donde N es
     igual al entero especificado con este parámetro, por ejemplo, 3.
# -t número de caracteres usados en el algoritmo Burrows-Wheeler para comprimir el
     tamaño de la lectura al juntar caracteres repetidos. En este caso está el
    valor por defecto, 10.
# -x este parámetro sirve para fijar un nombre base al genoma de referencia, el
    cual está en la variable REFERENCE.
# -1 parámetro que indica el archivo del primer conjunto de lecturas (forward).
# -2 parámetro que indica el archivo del segundo conjunto de lecturas (reverse).
# 2> instrucción para indicar que habrá un archivo .log en el que se describirá qu
    é se está ejecutando y cómo los argumentos anteriores están siendo evaluados.
# la salida de la ejecución de Bowtie2 es leída por SAMTOOLS.
# view es la instrucción para visualizar este tipo de archivos .sam.
# -b indica que el resultado sea transformado al formato .bam
# -h indica que el resultado incluya un encabezado.
# -S opcional, usado antes en versiones anteriores de Samtools cuando el archivo
    estaba en formato .sam y no en otros como .bam o .cram.

mkdir ${save_location}_tmpdir; #Crea un directorio temporal para guardar archivos
    que luego serán eliminados.

$JAVA -jar ${PICARD} SortSam MAX_RECORDS_IN_RAM=1000000 SO=coordinate CREATE_INDEX
    =true TMP_DIR=${save_location}_tmpdir I=${save_location}_bowtie2.bam O=${
    save_location}_bowtie2_sorted.bam >& ${save_location}_bowtie2_sort.log; #Picard
     y Java son usados para ordenar los .bam generados en el paso anterior con
    Bowtie2.
# Sortsam es una instrucción que indica el ordenamiento de un archivo .bam, en
    este caso, por coordenadas (SO=coordinate).
# MAX_RECORDS_IN_RAM indica el número de registros que se van a almacenar en RAM
    antes de escribir en disco. El valor por defecto es 5000000, pero en este caso
    se usó 1000000.
# CREATE_INDEX es un booleano para indicar si se crea un índice para el archivo .
    bam cuando se genera el archivo ordenado por coordenadas .bam
# TMP_DIR directorio para almacenar archivos temporal que luego serán eliminados.
# I es el archivo de entrada, los alineamientos sin ordenar.
# O es el archivo de salida, los alineamientos ordenados.
# >& instrucción para indicar que habrá un archivo .log en el que se describirá qu
    é se está ejecutando y cómo los argumentos (anteriores) están siendo evaluados.

rm -rf ${save_location}_tmpdir; #Elimina el directorio temporal con archivos que
    no tendrán más uso.
```

A.3. Llamado de variantes y filtros

A.3.1. LLamado de variantes a partir de los alineamientos a la referencia

Algoritmo A.4: instrucciones usando NGSEP para hacer el llamado de variantes a partir de los alineamientos a la referencia. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 04_findvariants_noknownvariants.sh
# Fecha: 21-09-2020
# Objetivo: Hacer un llamado de variantes inicial para crear un catálogo de las
    mismas.
# Salida: archivos .vcf para cada individuo.
```

```
# Entrada: archivos _sorted.bam por cada uno de los 95 individuos.


p=$1; #parámetro de entrada el cual es la ruta de ubicación del archivo de alineamiento odenado (sorted.bam) para un individuo.
sample=`echo ${p} | awk -F "/" '{print $7}' | sed 's/_USD.*//g'`; #extracción del id de un individuo a partir del parámetro $p.
save_location=../findvariants/${sample}/; #ubicación de la ruta donde se guardará el resultado del llamado de variantes sin un catálogo de las mismas.

REFERENCE=../reference/scaffolds_final_NGSEP_ polished.fa; # variable cuya asignación es la ruta del genoma de referencia, los scaffolds de CC-01-1940.

KNOWNSTRs=../mulvardet/scaffolds_final_ polished_trf_2_7_7_80_10_20_50.txt; # archivo con los microsatélites conocidos en el genoma: short tandem reapeats (STRs). Este es un archivo de texto con al menos tres columnas: cromosoma, primera posición y última posición. Las posiciones deben ser de base 1 e inclusivas. Leer tesis doctoral de Trujillo, 2020 para detallar cómo generar este archivo.

JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada para la ejecución de los procesos, en este caso 64 GB.

NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de variantes.

$JAVA -jar ${NGSEP} FindVariants -minQuality 40 -maxBaseQS 30 -knownSTRs ${KNOWNSTRs} -sampleId ${sample} -ploidy 10 -psp ${REFERENCE} ${p} ${save_location}findvariants.${sample} >& ${save_location}ngsep_findvariants.${sample}.log;

#Ejecución de NGSEP con la instrucción FindVariants:
# -minQuality mínima calidad de genotipo para aceptar una
# detección de SNV(Single Nucleotide Variation). En este caso es de 40.
# -maxBaseQS máximo valor permitido para puntaje de calidad de
# en una base (Escala Phred). En este caso es de 30.
# -knownSTRs parámetro para aceptar un archivo donde están registrados los microsatélites.
# -sampleId parámetro para pasar a NGSEP el nombre del id del individuo en la variable $sample.
# -ploidy parámetro para indicar a NGSEP cuál es la ploidía de las muestras. En este caso 10.
# -psp parámetro para indicar a NGSEP que debe poner un encabezado en el archivo .vcf de salida que ayuda para identificar la ploidía de las muestras en los análisis posteriores.
# ${save_location}findvariants.${sample} es el archivo de salida.
# >& instrucción para indicar que habrá un archivo ${save_location}ngsep_findvariants.${sample}.log en el que se describirá qué se está ejecutando y cómo los argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.2. Unión de variantes detectadas

Algoritmo A.5: instrucciones usando NGSEP para hacer la unión de variantes detectadas. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 05_mergevariants.sh
# Fecha: 21-09-2020
# Objetivo: hacer una unión de variantes que se descubre en cada individuo del paso anterior y se ejecuta una conciliación de alelos para las variantes intersectadas.
# Salida: archivo .vcf de toda la población.
# Entrada: archivos .vcf por cada uno de los 95 individuos generados a partir de 04_findvariants_noknownvariants.sh.

save_location=../results/mergevariants/;
# variable cuya asignación es la ruta de almacenamiento donde se guardará el archivo de salida.

load_location=../results/symbolic_links_findvariants/;
# variable cuya asignación es el directorio donde están todos los links simbólicos que apuntan a donde están los .vcf generados para cada unos de los individuos.

SEQUENCENAMES=../reference/scaffolds_final_NGSEP_ polished.fa.fai;
# variable cuya asignación es el archivo donde está el nombre de las secuencias en el genoma de referenica, en este caso, los scaffolds. El archivo de nombres de secuencias es un archivo de texto que sólo tiene los ids de las secuencias en la referencia. Es usado por el programa para determinar el orden de las secuencias de referencia. En el documento este archivo es referenciado como seq_names.txt.


JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada para la ejecución de los procesos, en este caso 64 GB.

NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de variantes.

$JAVA -jar ${NGSEP} MergeVariants ${SEQUENCENAMES} ${save_location}mergevariants.95ids.vcf ${load_location}*.vcf >& ${save_location}ngsep_mergevariants.95ids.log;
#Ejecución de NGSEP con la instrucción MergeVariants:
# $SEQUENCENAMES es la variable cuyo valor tiene la ruta donde está el archivo con el nombre de la secuencias.
# ${save_location}mergevariants.95ids.vcf ruta y nombre del archivo donde se guardará el  .vcf resultante.
# ${load_location}*.vcf ruta donde están todos los links simbólicos que apuntan a la dirrección donde están los .vcf resultantes de cada individuo del paso FindVariants.
# >& instrucción para indicar que habrá un archivo ${save_location}ngsep_mergevariants.95ids.log en el que se describirá qué se está ejecutando y cómo los argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.3. Colección de *scaffolds* pequeños y regiones repetitivas

Algoritmo A.6: instrucciones usando awk para coleccionar *scaffolds* y regiones repetitivas no útiles para la detección de SNPs.

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 06_1_filtervcf_mergevariants.sh
# Fecha: 21-09-2020
# Objetivo: seleccionar y almacenar en un archivo los scaffolds que son menores a 100 kbp y no son pseudocromosomas o tienen regiones repetitivas.
# Salida: archivo lessthan100kbAND_NOTpseudochromosomes_scaffoldsORrepetitive_regions.txt que tiene tres columnas: el id de la secuencia o scaffold, número en el que inicia la secuencia y número en el que termina la secuencia.
# Entrada: tres archivos: pseudochromosome_scaffolds.txt (listado de scaffolds que son pseudocromosomas (revisar Trujillo, 2020 para entender cómo producir este archivo)), scaffolds_final_NGSEP_polished.fa.fai (ver final de este archivo) y scaffolds_final_NGSEP_polished_repeatmasker.fa.out (listado de regiones repetitivas dentro del conjunto de scaffolds (revisar Trujillo, 2020 para entender cómo producir este archivo))

grep -v -w -f pseudochromosome_scaffolds.txt scaffolds_final_NGSEP_polished.fa.fai | \ # El grep -v invierte la selección para arrojar lo contrario a lo que se está buscando, -w escoge que el match esté como una sola palabra y no dentro de una cadena de texto y el -f es para que reconozca el archivo pseudochromosome_scaffolds.txt como un listado de patrones para buscar (el cual es la lista de pseudocromosomas).
 awk '{if ($2 < 100000) print $1 "\t1\t" $2}' > lessthan100kbAND_NOTpseudochromosomes_scaffoldsORrepetitive_regions.txt # Después del pipe se busca que la segunda columna por scaffold no exceda el valor 100000 porque un scaffold de menor tamaño se considera como pequeño. Finalmente se imprime el nombre de la secuencia ($1), el 1 (uno) porque es todo un scaffold desde la primera pb hasta la última y el tamaño del scaffold (última pb).

awk '{if (NR > 3) print $5 "\t" $6 "\t" $7}' scaffolds_final_NGSEP_polished_repeatmasker.fa.out >> lessthan100kbAND_NOTpseudochromosomes_scaffoldsORrepetitive_regions.txt # Esta instrucción busca añadir al archivo creado en la instrucción anterior los scaffolds que tienen regiones repetitivas. Desde el archivo de scaffolds_final_NGSEP_polished_repeatmasker.fa.out se imprime las columnas $5, $6 y $7 que corresponden respectivamente al nombre del scaffold, en qué base (pb) inicia la región y en qué base termina la región repetitiva. Esta impresión se hace después de la línea 3 porque antes hay un encabezado en el archivo.


# el archivo scaffolds_final_NGSEP_polished.fa.fai se produce de la siguiente forma:
:'
    $ samtools faidx scaffolds_final_NGSEP_polished.fa
'
# Esta instrucción genera un índice de referencia de las secuencias enlistadas en el archivo de entrada .fa (formato FASTA). El archivo de salida .fa.fai es uno que contiene cinco columnas: id de la secuencia, tamaño de la secuencia, número de caracteres en el que inicia la primera base (pb) de esa secuencia, número de caracteres o pbs que hay en cada línea y número de caracteres o pbs que hay en cada línea contando el salto de línea.
```

A.3.4. Filtro de regiones repetitivas y *scaffolds* pequeños

Algoritmo A.7: instrucciones usando NGSEP para descartar *scaffolds* y regiones repetitivas no útiles para la detección de SNPs. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 06_2_filtervcf_mergevariants.sh
# Fecha: 21-09-2020
# Objetivo: hacer un filtro al archivo resultante del MergeVariants dado el gran conjunto de datos para hacer una reducción de variantes y acelerar el proceso.
# Salida: archivo .vcf de toda la población con una reducción de loci.
# Entrada: archivo .vcf resultante del código 05_mergevariants.sh.


save_location=../results/mergevariants/;
# variable cuya asignación es la ruta de almacenamiento donde se guardará el archivo de salida.

VCF_in=../mergevariants/mergevariants_ filtered.95ids.vcf;
# variable cuya asignación es la ruta de almacenamiento donde está guardado el archivo que resultó de ejecutar el script 05_mergevariants.sh.

FILTER_GENOMICREGIONS=../reference/lessthan100kbAND_NOTpseudochromosomes_scaffoldsORrepetitive_regions.txt;

# variable cuya asignación es la ruta de almacenamiento donde está guardado el archivo que tiene scaffolds con tamaño menor a 100kb, que no fueran pseudocromosomas y regiones repetitivas. Resultado del algoritmo A.6.

JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada para la ejecución de los procesos, en este caso 64 GB.

NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de variantes.


${JAVA} -jar ${NGSEP} FilterVCF -d 20 -s -frs ${FILTER_GENOMICREGIONS} ${VCF_in} 1> ${save_location}mergevariants_filtered-d20added.95ids.vcf 2> ${save_location}ngsep_mergevariants_filtered-d20added.95ids.log;

#Ejecución de NGSEP con la instrucción FilterVCF:
# -d es el parámetro usado para indicar cuantas pares de bases de distancia tienen que haber entre cada par de loci. En este caso fueron 20.
# -s es un parámetro para seleccionar solo los SNPs bialélicos.
# -frs parámetro para indicar que el archivo ${FILTER_GENOMICREGIONS}, que tiene regiones genómicas, no sea considerado.
# VCF_in es la variable asignada anteriormente con la dirección donde está el .vcf de entrada.
# 1> ${save_location}mergevariants_filtered-d20added.95ids.vcf nombre del archivo donde quedará guardado el resultado de la ejecución de FilterVCF.
# 2> instrucción para indicar que habrá un archivo ${save_location}ngsep_mergevariants_filtered-d20added.95ids.log en el que se describirá qué se está ejecutando y cómo los argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.5. Genotipificación

Algoritmo A.8: instrucciones usando NGSEP para hacer la genotipificación de los marcadores (heterocigotos u homocigotos) en cada uno de los individuos. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 07_findvariants_knownvariants.sh
# Fecha: 21-09-2020
# Objetivo: hacer una genotipificación de los SNPs enlistados y descubiertos en los pasos anteriores para saber si cada individuo es heterocigoto, homocigoto al alelo de referencia u homocigoto al alelo alternativo para un loci.
# Salida: archivos .vcf para cada individuo.
# Entrada: archivos _sorted.bam por cada uno de los 95 individuos.

p=$1; #parámetro de entrada el cual es la ruta de ubicación del archivo de alineamiento odenado (sorted.bam) para un individuo.
sample=`echo ${p} | awk -F "/" '{print $7}' |sed 's/_USD.*//g'`; #extracción del id de un individuo a partir del parámetro $p.
save_location=/bioinfotmp2/genetic_map_CC_01_1940/results/findvariants/${sample}/; #ubicación de la ruta donde se guardará el resultado del llamado de variantes con un catálogo de las mismas.

REFERENCE=../reference/scaffolds_final_NGSEP_ polished.fa; # variable cuya asignación es la ruta del genoma de referencia, los scaffolds de CC-01-1940.

KNOWNVARIANTS=../mergevariants/mergevariants_ filtered-d20added.95ids.vcf; # variable cuya asignación es la ruta del catálogo de las variantes, archivo resultante de la ejecución de 06_filtervcf_mergevariants.

JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada para la ejecución de los procesos, en este caso 64 GB.

NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de variantes.


# Running Java
$JAVA -jar ${NGSEP} FindVariants -minQuality 40 -maxBaseQS 30 -knownVariants ${KNOWNVARIANTS} -sampleId ${sample} -ploidy 10 -psp ${REFERENCE} ${p} ${save_location}findvariants_knownvariants.${sample} >& ${save_location}ngsep_findvariants_knownvariants.${sample}.log;

#Ejecución de NGSEP con la instrucción FindVariants:
# -minQuality mínima calidad de genotipo para aceptar una
# detección de SNV(Single Nucleotide Variation). En este caso es de 40.
# -maxBaseQS máximo valor permitido para puntaje de calidad de
# en una base (Escala Phred). En este caso es de 30.
# -knownVariants parámetro para aceptar un archivo donde está registrado el catálogo de variantes.
# -sampleId parámetro para pasar a NGSEP el nombre del id del individuo en la variable $sample.
# -ploidy parámetro para indicar a NGSEP cuál es la ploidía de las muestras. En este caso 10.
# -psp parámetro para indicar a NGSEP que debe poner un encabezado en el archivo .vcf de salida que ayuda para identificar la ploidía de las muestras en los aná
```

```
    lisis posteriores.
# ${save_location}findvariants_knownvariants.${sample} es el archivo de salida.
# >& instrucción para indicar que habrá un archivo ${save_location}
    ngsep_findvariants.${sample}.log en el que se describirá qué se está ejecutando
     y cómo los argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier
    inconsistencia por el uso de versiones, es mejor revisar la documentación
    oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.6. Unión del genotipado por cada individuo

Algoritmo A.9: instrucciones usando NGSEP para hacer una unión del genotipado por cada individuo y obtener el `.vcf` previo al inicio de la fase de filtros. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 08_mergevcf.sh
# Fecha: 21-09-2020
# Objetivo: hacer una mezcla y consolidar todos los datos genotipados de la
    población en un único archivo
# Salida: archivo .vcf de toda la población.
# Entrada: archivos .vcf por cada uno de los 95 individuos generados a partir de
    07_findvariants_knownvariants.sh.


save_location=../results/mergevcf/;
# variable cuya asignación es la ruta de almacenamiento donde se guardará el
    archivo de salida.

load_location=../results/symbolic_links_findvariants_ knownvariants/;
# variable cuya asignación es el directorio donde están todos los links simbólicos
     que apuntan a donde están los .vcf generados para cada unos de los individuos.

SEQUENCENAMES=../reference/scaffolds_final_NGSEP_polished.fa.fai;
# variable cuya asignación es el archivo donde está el nombre de las secuencias en
     el genoma de referenica, en este caso, los scaffolds. El archivo de nombres de
     secuencias es un archivo de texto que sólo tiene los ids de las secuencias en
    la referencia. Es usado por el programa para determinar el orden de las
    secuencias de referencia. En el documento este archivo es referenciado como
    seq_names.txt.

JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada
    para la ejecución de los procesos, en este caso 64 GB.


NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de
    variantes.

$JAVA -jar ${NGSEP} MergeVCF ${SEQUENCENAMES} ${load_location}*.vcf 1> ${
    save_location}mergevcf.95ids.vcf 2> ${save_location}ngsep_mergevcf.95ids.log;
# Ejecución de NGSEP con la instrucción MergeVCF:
# $SEQUENCENAMES es la variable cuyo valor tiene la ruta donde está el archivo con
     el nombre de la secuencias.
# ${load_location}*.vcf ruta donde están todos los links simbólicos que apuntan a
    la dirrección donde están los .vcf resultantes de cada individuo del paso
    FindVariants.
```

```
# ${save_location}mergevcf.95ids.vcf ruta y nombre del archivo donde se guardará el  .vcf resultante.
# 2> instrucción para indicar que habrá un archivo ${save_location}ngsep_mergevcf.95ids.log en el que se describirá qué se está ejecutando y cómo los argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.7. Distribución de OH y MAF para evaluar filtros

Algoritmo A.10: Algoritmo para extraer la distribución de la heterocigosidad observada en un conjunto de loci provenientes de un archivo `.vcf`. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: oh_distribution.sh
# Fecha: 21-09-2020
# Objetivo: calcular la distribución de la heterocigosidad observada en la población según los datos calculados con la instrucción DiversityStats de NGSEP (ver instrucción anexa al final del algoritmo).
# Salida: archivo de texto con dos columnas: la primera es el valor de la heterocigosidad de 0.0 a 1.0 y la segunda columna es la cantidad de SNPs que tienen esa heterocigosidad.
# Entrada: archivo de DiversityStats calculado con NGSEP. En este archivo la 3 columna debe ser la correspondiente a la heterocigosidad observada.

save_location=../mergevcf/observed_heterozygosity.95ids.log;
load_location=../mergevcf/diversitystats_mergevcf.95ids.log;

#OH

awk -F ':' '
  m=($3 ~ /[0-9]/) {print $3}
' ${load_location} |
sort |
awk -v s=0 -v e=1.0 -v d=0.01 '
  BEGIN { m = 1/d }
  { a[int($1*m)]++ }
  END{ e *= m; for(s = int(s*m); s <= e; s++) print s*d, a[s]+0 }
' > ${save_location}

# Esta instrucción toma cada heterocigosidad observada de cada loci del archivo generado de DiversityStats y cuenta cuantos loci hay en un rango de 0.01. Por ejemplo, cuenta cuantos loci hay de 0.10 a 0.11, luego de 0.11 a 0.12 y así sucesivamente hasta 1.0. La s (start) es el inicio de la distribución, la e (end) es el final de la distribución, la d (delta) es el cambio para definir un rango y la m (mark) es un arreglo computacional dado que generar rangos al repetir una sumatoria de 0.01 no está bien porque son número flotantes, no pueden ser representados en base 2 por lo que el error se acumula cada vez que se suma. Este arreglo se le hace a 's', a 'e' y a cada valor leído del archivo de entrada ($1*m).
# Este programa lo que hace es crear un arreglo asociativo. Asumamos que la entrada (después del sort) es:
:'
    0.2
    0.2
    0.2
    0.2
```

```
    0.2
    0.2
    0.2024
    0.2025
    0.2027
    0.2027
    0.2029
    0.2059
    0.2059
    0.2059
    0.2059
    0.2099
    0.2099
    0.2099
    0.2105
    0.2113
    0.2113
    0.2195
    0.2198
    0.2206
    0.2206
    0.2206
    0.2989
    0.2989
    0.2989
    0.3
    0.3

'
# Nuestro arreglo asociativo creado por a[int($1*m)]++ (++ suma el valor donde est
    á guardado, p. ej. la llave 200. Es decir, en cada iteración a[200] suma uno a
    su valor almacenado: a[200] = 1, a[200] = 2, ... Si $1 no ha cambiado) quedaría
     de esta forma:

:'
    20 18
    21 5
    22 3
    23 0
    24 0
    25 0
    26 0
    27 0
    28 0
    29 3
    30 2
'
# Al final, después de la instrucción END, se recorre este arreglo para imprimirlo
     y en la impresión transformamos 20 a 0.20, 21 a 0.21, ... con la operación s*d
     e imprimimos el contenido (a[s], aquí se suma 0 para asegurar que sea un nú
    mero y no otro tipo de dato). Finalmente, el script nos entregaría esto:

:'
    0.2 18
    0.21 5
    0.22 3
    0.23 0
    0.24 0
    0.25 0
```

```
    0.26 0
    0.27 0
    0.28 0
    0.29 3
    0.3 2
'

# instrucción para generar el archivo de diversity stats.

:'
    java -jar NGSEPcore_4.0.1.jar VCFDiversityStats -i mergevcf.95ids.vcf -o
    diversitystats_mergevcf.95ids.log
'

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier
    inconsistencia por el uso de versiones, es mejor revisar la documentación
    oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

Algoritmo A.11: Algoritmo para extraer la distribución de la frecuencia del alelo menor en un conjunto de loci provenientes de un archivo `.vcf`. [6]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: maf_distribution.sh
# Fecha: 21-09-2020
# Objetivo: calcular la distribución de la frecuencia del alelo menor (MAF) en la
    población según los datos calculados con la instrucción DiversityStats de NGSEP
     (ver instrucción anexa al final del algoritmo).
# Salida: archivo de texto con dos columnas: la primera es el rango del MAF de 0.0
     a 0.5 y la segunda columna es la cantidad de SNPs que tienen ese MAF.
# Entrada: archivo de DiversityStats calculado con NGSEP. En este archivo la 5
    columna debe ser la correspondiente al MAF.

save_location=../mergevcf/minor_allele_frequency.95ids.log;
load_location=../mergevcf/diversitystats_mergevcf.95ids.log;

# MAF

awk -F ':' '
  m=($5 ~ /[0-9]/) {print $5}
' ${load_location} |
sort |
awk -v s=0 -v e=0.5 -v d=0.01 '
  BEGIN { m = 1/d }
  { a[int($1*m)]++ }
  END{ e *= m; for(s = int(s*m); s <= e; s++){
      if (s == e)
        print s*d, a[s]+0
      else
        print s*d"-"(s+1)*d, a[s]+0
    }
  }
' > ${save_location}


# Este algoritmo, en general, sigue el mismo principio explicado que en el
    algoritmo para calcular la distribución de la OH, revisar esa explicación. Aquí
     la diferencia es que imprimimos un rango "0-0.01, 0.01-0.02, ..." en la
    primera columna. Esto sucede desde 0 hasta 0.49-0.5. En 0.5 no se imprime el
    rango y por eso es que se hace la condición "s == e", solo imprime 0.5 y el
```

```
    valor almacenado en a[s].
31
32
33 # instrucción para generar el archivo de diversity stats.
34
35 :'
36     java -jar NGSEPcore_4.0.1.jar VCFDiversityStats -i mergevcf.95ids.vcf -o
    diversitystats_mergevcf.95ids.log
37 '
38
39 # Esta documentación está basada en la versión usada de NGSEP, así que cualquier
    inconsistencia por el uso de versiones, es mejor revisar la documentación
    oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.8. Primer filtro de OH y MAF

Algoritmo A.12: instrucciones usando NGSEP para hacer el primer filtro de heterocigosis observada (OH) y frecuencia del alelo menor (MAF). [6,7]

```
1 # Autor(es): Giann Karlo Aguirre Samboní.
2 # Nombre del programa: 09_filtervcf_firstfilter.sh
3 # Fecha: 21-09-2020
4 # Objetivo: aplicar el primer filtro en el archivo .vcf donde están los loci
    genotipados para cada individuo. Este filtro tiene que ver con el MAF y la OH.
5 # Salida: archivo .vcf de toda la población con solamente los loci que cumplen los
     filtros.
6 # Entrada: archivo .vcf resultado de la ejecución de 08_mergevcf.sh.
7
8 save_location=../results/mergevcf/;
9 # variable cuya asignación es la ruta de almacenamiento donde se guardará el
    archivo de salida.
10
11 VCF_in=../mergevcf/mergevcf.95ids.vcf;
12 # variable cuya asignación es la ruta de almacenamiento donde está guardado el
    archivo que resultó de ejecutar el script 08_mergevcf.sh.
13
14 JAVA="java -d64 -Xmx64g";
15 #variable JAVA para asignar parámetros por defecto para la ejecución de Java.
16 # -d64 indica que la ejecución se haga en un ambiente de 64-bits.
17 # -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada
    para la ejecución de los procesos, en este caso 64 GB.
18
19 NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de
    variantes.
20
21 ${JAVA} -jar ${NGSEP} FilterVCF -minI 60 -minMAF 0.10 -minOH 0.10 -maxOH 0.90 ${
    VCF_in} 1> ${save_location}mergevcf.95ids.b_firstfiltered.vcf 2> ${
    save_location}mergevcf.95ids.b_firstfiltered.log;
22
23 #Ejecución de NGSEP con la instrucción FilterVCF:
24 # -minI es el parámetro usado para indicar cuantos individuos debe haber
    genotipados en el loci para no ser descartado. En este caso son 60.
25 # -minMAF es un parámetro para indicar cuál es el MAF mínimo que debe tener el
    loci para no ser descartado. En este caso es 0.10.
26 # -minOH es un parámetro para indicar cuál es la OH mínima que debe tener el loci
    para no ser descartado. En este caso es 0.10.
27 # -maxOH es un parámetro para indicar cuál es la OH máxima que debe tener el loci
    para no ser descartado. En este caso es 0.90.
```

```
# VCF_in es la variable asignada anteriormente con la dirección donde está el .vcf de entrada.
# 1> ${save_location}mergevcf.95ids.b_firstfiltered.vcf nombre del archivo donde quedará guardado el resultado de la ejecución de FilterVCF.
# 2> instrucción para indicar que habrá un archivo ${save_location}mergevcf.95ids.b_firstfiltered.log en el que se describirá qué se está ejecutando y cómo los argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.9. Filtro de datos pérdidos u homocigotos en los parentales

Algoritmo A.13: instrucciones usando awk para hacer el filtro de datos pérdidos o cuando el alelo es homocigoto al mismo alelo en ambos padres. [6,7,24]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 10_filter_homozygous_parents_secondfilter.sh
# Fecha: 21-09-2020
# Objetivo: descartar aquellos loci que no tienen genotipificación en los padres o que sean homocigotos (al alelo de referencia o al alelo alternativo).
# Salida: archivo .vcf de toda la población con solamente los loci que cumplen los filtros.
# Entrada: archivo .vcf resultado de la ejecución de 09_filtervcf_firstfilter.sh.

awk '
{
  if($1 ~ /#/)
    print $0
  else if (!($10 ~ "1/1" && $11 ~ "1/1"))
      if (!($10 ~ "0/0" && $11 ~ "0/0"))
        if ($10 !~ "\\./\\.")
          if ($11 !~ "\\./\\.")
            print $0
}' ../results/mergevcf/mergevcf.95ids.b_firstfiltered.vcf > ../results/mergevcf/mergevcf.95ids.b_secondfiltered.vcf
```

A.3.10. Segundo filtro de OH y MAF

Algoritmo A.14: instrucciones usando NGSEP para hacer el segundo filtro de heterocigosis observada (OH) y frecuencia del alelo menor (MAF). [7]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 11_filtervcf_thirdfilter.sh
# Fecha: 21-09-2020
# Objetivo: aplicar el tercer filtro en el archivo .vcf donde están los loci genotipados para cada individuo. Este filtro tiene que ver con el MAF y la OH, es más estricto que los rangos en 09_filtervcf_firstfilter.sh.
# Salida: archivo .vcf de toda la población con solamente los loci que cumplen los filtros.
# Entrada: archivo .vcf resultado de la ejecución de 10_filter_homozygous_parents_secondfilter.sh.

save_location=../results/mergevcf/;
# variable cuya asignación es la ruta de almacenamiento donde se guardará el archivo de salida.

```

```
VCF_in=../mergevcf/mergevcf.95ids. b_secondfiltered.vcf;
# variable cuya asignación es la ruta de almacenamiento donde está guardado el
    archivo que resultó de ejecutar el script 10
    _filter_homozygous_parents_secondfilter.sh.

JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada
    para la ejecución de los procesos, en este caso 64 GB.

NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de
    variantes.


${JAVA} -jar ${NGSEP} FilterVCF -minMAF 0.20 -maxMAF 0.40 -minOH 0.35 -maxOH 0.65
    ${VCF_in} 1> ${save_location}mergevcf.95ids.c_thirdfiltered.vcf 2> ${
    save_location}mergevcf.95ids.c_thirdfiltered.log;

#Ejecución de NGSEP con la instrucción FilterVCF:
# -minMAF es un parámetro para indicar cuál es el MAF mínimo que debe tener el
    loci para no ser descartado. En este caso es 0.20.
# -maxMAF es un parámetro para indicar cuál es el MAF mínimo que debe tener el
    loci para no ser descartado. En este caso es 0.40.
# -minOH es un parámetro para indicar cuál es la OH mínima que debe tener el loci
    para no ser descartado. En este caso es 0.35.
# -maxOH es un parámetro para indicar cuál es la OH máxima que debe tener el loci
    para no ser descartado. En este caso es 0.65.
# VCF_in es la variable asignada anteriormente con la dirección donde está el .vcf
     de entrada.
# 1> ${save_location}mergevcf.95ids.c_thirdfiltered.vcf nombre del archivo donde
    quedará guardado el resultado de la ejecución de FilterVCF.
# 2> instrucción para indicar que habrá un archivo ${save_location}mergevcf.95ids.
    c_thirdfiltered.log en el que se describirá qué se está ejecutando y cómo los
    argumentos anteriores están siendo evaluados.

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier
    inconsistencia por el uso de versiones, es mejor revisar la documentación
    oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.11. SNPs sin desviación significativa de HWE

Algoritmo A.15: instrucciones usando awk para para colectar aquellos SNPs que no se desvían significativamente del HWE. [7,24]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 12_nosignificance_deviation.sh
# Fecha: 21-09-2020
# Objetivo: seleccionar aquellos loci que no tienen una desviación significativa
    de HWE. Esto es cuando el valor p es mayor a 0.01.
# Salida: archivo .vcf de toda la población con solamente los loci que cumplen el
    filtro.
# Entrada: archivo .vcf resultado de la ejecución de 11_filtervcf_thirdfilter.sh.

awk -F: '
{
  if($8 > 0.01){
    print $0
  }
```

```
}' ../results/mergevcf/diversitystats_mergevcf .95ids.b_secondfiltered.log | awk '
    { print $1,$2,$3 }' > ../results/mergevcf/snps_with_hwe_diversitystats_
    mergevcf.95ids.001b_secondfiltered.log
```

A.3.12. Seleccionar SNPs en HWE

Algoritmo A.16: instrucciones usando NGSEP para hacer el filtro de seleccionar solo aquellos SNPs no desviados significativamente del HWE. [7]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 11_filtervcf_thirdfilter.sh
# Fecha: 21-09-2020
# Objetivo: aplicar el cuarto filtro en el archivo .vcf donde están los loci
    genotipados para cada individuo. Este filtro tiene que ver con descartar los
    loci que se desvían significativamente del HWE.
# Salida: archivo .vcf de toda la población con solamente los loci que cumplen el
    filtro.
# Entrada: archivo .vcf resultado de la ejecución de 11_filtervcf_thirdfilter.sh.

save_location=../results/mergevcf/;
# variable cuya asignación es la ruta de almacenamiento donde se guardará el
    archivo de salida.

VCF_in=../results/mergevcf/mergevcf.95ids. c_thirdfiltered.vcf;
# variable cuya asignación es la ruta de almacenamiento donde está guardado el
    archivo que resultó de ejecutar el script 10
    _filter_homozygous_parents_secondfilter.sh.

SelectSNPs=../results/mergevcf/snps_with_hwe_ diversitystats_mergevcf.95ids.001
    b_secondfiltered.log;
# archivo donde están registrados los ids de los SNPs que no se desvían
    significativamente del HWE.

JAVA="java -d64 -Xmx64g";
#variable JAVA para asignar parámetros por defecto para la ejecución de Java.
# -d64 indica que la ejecución se haga en un ambiente de 64-bits.
# -Xmx64g (igual a -XX:MaxHeapSize = 64g) indica la cantidad de memoria destinada
    para la ejecución de los procesos, en este caso 64 GB.

NGSEP=../ngsep_lib/NGSEPcore_3.3.3.jar; #módulo de Java para hacer el llamado de
    variantes.

${JAVA} -jar ${NGSEP} FilterVCF -srs snps_with_hwe_diversitystats_mergevcf.95ids
    .001b_secondfiltered.log ${VCF_in} 1> ${save_location}mergevcf.95ids.
    c_fourthfiltered.vcf 2> ${save_location}mergevcf.95ids.c_fourthfiltered.log;

#Ejecución de NGSEP con la instrucción FilterVCF:
# -srs es un parámetro para indicar que el archivo
    snps_with_hwe_diversitystats_mergevcf.95ids.001b_secondfiltered.log debe ser
    considerado para seleccionar aquellos SNPs que están ahí escritos.
# VCF_in es la variable asignada anteriormente con la dirección donde está el .vcf
     de entrada.
# 1> ${save_location}mergevcf.95ids.c_fourthfiltered.vcf nombre del archivo donde
    quedará guardado el resultado de la ejecución de FilterVCF.
# 2> instrucción para indicar que habrá un archivo ${save_location}mergevcf.95ids.
    c_fourthfiltered.log en el que se describirá qué se está ejecutando y cómo los
    argumentos anteriores están siendo evaluados.

```

```
# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.3.13. Loci heterocigotos para el primer parental

Algoritmo A.17: comandos aplicados en la estrategia C cuyos loci filtrados son de interés para construir el mapa genético del parental 1, CC 01-1940. [7,25]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 15_runp1_genetic_map.sh
# Fecha: 21-09-2020
# Objetivo: filtrar los marcadores en un vcf que son heterocigotos para el primer parental (0/1) pero homocigotos al alelo de referencia (0/0) o al alelo alternativo (1/1) para el segundo parental.
# Salida: archivo .vcf filtrado según el objetivo.
# Entrada: archivo .vcf sin filtrar con el genotipado de la población

awk '
{
  if($1 ~ /#/)
    print $0
  else if ($10 ~ "0/1" && ($11 ~ "0/0" || $11 ~ "1/1"))
     print $0
}' ../mergevcf/mergevcf.95ids.vcf > ../mergevcf/mergevcf.95ids.p1.d_firstfiltered.vcf

# En script se usa el programa awk.
# En el if se indica que todas las líneas que contengan el caracter numeral sean impresas. De lo contrario, si la columna 10 (parental 1) para ese marcador tiene un genotipo heterocigoto y para el segundo parental (columna 11) el genotipo es homocigoto (a la referencia o alternativo), entonces imprima toda la línea. De esta manera se descartan los loci que no cumplan la condición.
```

A.3.14. Conteo de genotipos y datos perdidos

Algoritmo A.18: comandos aplicados en la estrategia C en los que se cuentan los genotipos y los datos perdidos (*misdata*) por cada loci en archivo `.vcf`. [7,25]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 16_counting_genotypes_misdata.sh
# Fecha: 21-09-2020
# Objetivo: contar el número de genotipos heterocigotos, homocigotos y datos perdidos para sacar un total de datos que se tengan por loci. Es importante notar que el parental está en la posición (columna) 10 del archivo (i = 10), si se usa otro vcf con el primer individuo en otra posición, habrá que cambiar este valor.
# Salida: archivo de texto con cuatro columnas: número de heterocigotos, homocigotos, misdata y el total (heterocigotos + homocigotos). Estos cuatro valores se dan por cada loci.
# Entrada: archivo .vcf resultado de aquellos loci que sean heterocigotos para el primer parental pero homocigotos para el segundo(para este caso sería el que resultó del script runp1_genetic_map.sh).

awk -v het=0 -v hom=0 -v mis=0 -v tot=0 '
{
  if ($1 !~ /#/)
  {
```

```
    i = 10
    while(i<=NF){
      if ($i ~ "0/1")
        het += 1;
      else if ($i ~ "0/0" || $i ~ "1/1")
        hom += 1;
      else{
        mis += 1;
      }
      i += 1;
    }
    tot = het + hom;
    print het,hom,mis,tot;
    het = 0;
    hom = 0;
    mis = 0;
    tot = 0;
  }
}' ../mergevcf/mergevcf.95ids.p1.d_firstfiltered.vcf > ../mergevcf/mergevcf.95ids.
    p1.d_firstfiltered_miscount.log

# En script se usa el programa awk.
# Antes de iniciar el programa, en la línea 8 se crean 4 variables (het, hom, mis
    y tot). Cuando inicia el programa, se hace una condición que se cumple cuando
    en la primera columna del archivo no hay un '#'. Dentro de la condición se crea
     una variable i = 10 que indica la posición en la columna 10 donde está el
    primer individuo de la población, este caso el parental CC_011940.

# Después se hace un ciclo de i hasta NF (inclusive (NF es una variable reservada
    de awk para indicar el número total de columnas (se considera columna cuando
    hay un espacio en blanco) que hay en el archivo)). Dentro del ciclo se revisa
    si cada columna tiene un genotipo heterocigoto (0/1), homocigoto (0/0 o 1/1) o
    un dato perdido y se suma 1 respectivamente en het, hom o mis.

# Cuando termina el ciclo se calcula tot al sumar het + hom y se imprime en el
    archivo de salida las cuatro columnas: het, hom, mis y tot. Finalmente, se
    reinician las variables en 0 para calcularlas nuevamente en el siguiente loci.
```

A.3.15. Hoja de cálculo ejemplo para hacer una prueba chi cuadrado

Tabla A.3: muestra de ejemplo de lo que sería la hoja de cálculo útil para calcular el valor-p en un *software* como *Microsoft Excel*. [8]

obshet	obshom	misdata	total (obshet+obshom)	exphet	exphom	pvalue
26	52	17	78	42.5	42.5	0.0034945
38	5	52	43	42.5	42.5	0.0000000
31	54	10	85	42.5	42.5	0.0126064
56	26	13	82	42.5	42.5	0.0010748
30	58	7	88	42.5	42.5	0.0022550
23	55	17	78	42.5	42.5	0.0003809
21	60	14	81	42.5	42.5	0.0000212
7	78	10	85	42.5	42.5	0.0000000
13	74	8	87	42.5	42.5	0.0000000
7	73	15	80	42.5	42.5	0.0000000
20	62	13	82	42.5	42.5	0.0000049
48	33	14	81	42.5	42.5	0.0922139
33	51	11	84	42.5	42.5	0.0505377
28	63	4	91	42.5	42.5	0.0001173
34	49	12	83	42.5	42.5	0.1007192
53	29	13	82	42.5	42.5	0.0087051
11	79	5	90	42.5	42.5	0.0000000
7	80	8	87	42.5	42.5	0.0000000
18	64	13	82	42.5	42.5	0.0000006
43	42	10	85	42.5	42.5	0.9136267
1	85	9	86	42.5	42.5	0.0000000

A.3.16. Sesgo para heterocigotos

Algoritmo A.19: comandos aplicados en la estrategia C en los que se verifica por cada loci que el valor P supere un umbral para ser seleccionado en el `.vcf` resultante. [8,25]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 17_heterozygous_bias.sh
# Fecha: 21-09-2020
# Objetivo: incluir solo aquellos loci que tienen cierta desviación o sesgo con respecto a lo que se esperaría de un marcador de dosis única en una población biparental.
# Salida: archivo .vcf que cumplen con la condición de tener un valor-p mayor o igual a 0.1 (en este caso).
# Entrada: archivo producido del conteo de genotipos y misdata (con una adición de tres columnas (uso hoja de cálculo) correspondientes a valores esperados para heterocigotos, homocigotos y el valor-p), y archivo .vcf de la población.

awk -v loci=116 '
  FNR==NR{
    a[FNR+loci]=$7;
    next
  }
```

```
13   {
14     if($1 ~ /#/ || a[FNR] >= 0.1)
15       print $0;
16   }
17 ' ../mergevcf/mergevcf.95ids.p1.d_firstfiltered_miscount.log ../mergevcf/mergevcf
     .95ids.p1.d_firstfiltered.vcf > ../mergevcf/mergevcf.95ids.p1.d_secondfiltered.
     vcf
18
19 # basado en el valor-p, calculado con el archivo generado del conteo de genotipos
     y misdata, y con la ayuda de una hoja de cálculo para hacer un chi square test,
      se revisa que dicho valor supere un umbral. Se usa awk para lograr este
     objetivo.
20 # Antes de empezar el programa se crea una variable loci asignada al número 116.
     En este script se abren dos archivo al mismo tiempo (*miscount.log y *
     d_firstfiltered.vcf). FNR mantiener la lectura de línea por línea por archivo
     separado mientras que NR mantiene el conteo de líneas en los dos archivos, es
     decir que meintras FNR es igual a 0 cada vez que lee inicialmente un archivo,
     NR mantiene el acumulado de líneas. Con esto en mente, la condición FNR==NR
     indica que se cumpla solo cuando estamos leyendo el primer archivo. Cuando se
     cumple esta condición se crea un arreglo asociativo donde el índice es valor de
      FNR+loci (116 en este caso porque es el total de líneas que corresponden a los
      archivos .vcf que estamos manipulando), y el elemento corresponde al valor-p
     ya calculado ($7). La palabra reservada next indica que ejecute esa condición
     hasta que llegue al final del primer archivo.
21 # Cuando hemos terminado de leer todo el primer archivo y tenemos nuestro arreglo
     asociativo creado, leemos el segundo archivo. En este caso estamos leyendo un
     archivo .vcf entonces la condición dice que imprima en el archivo de salida .
     vcf la información que corresponde a un loci, solo en el caso de que la primera
      columna ($1) tenga un '#' (corresponde al encabezado) o que a[FNR] (ya hemos
     leído las primeras 116 líneas entonces FNR > 116) sea mayor o igual 0.1 (el
     sesgo que hemos permitido aceptar).
```

A.4. Mapa genético

A.4.1. Búsqueda de *softwares*

Tabla A.4: Palabras clave usadas en Google Scholar para establecer el puntaje del parámetro # Citas y usos. [8]

Google scholar	**Búsqueda o fuente**	**Resultados**	**Total**
JoinMap	Citas en la página web	1,060	2,760
	"joinmap software"	463	
	"joinmap package"	17	
	"joinmap program"	1,220	
Pergola	Citas en página web	0	5
	"pergola software"	1	
	"pergola package"	4	
	"pergola program"	0	
Netgwas	Citas en página web	0	6
	"netgwas software"	0	
	"netgwas package"	6	
	"netgwas program"	0	
OneMap	Citas en página web	0	349
	"OneMap software"	288	
	"OneMap package"	54	
	"OneMap program"	7	
polymapR	Citas en página web	0	22
	"polymapR software"	3	
	"polymapR package"	10	
	"polymapR program"	9	

Tabla A.5: Palabras clave usadas en Google Scholar para establecer el puntaje del parámetro Trabajos relacionados con caña. [8]

Google scholar	**Búsqueda o fuente**	**Resultados**	**Total**
JoinMap	Citas en la página web	2	1,110
	"joinmap software" + sugarcane	36	
	"joinmap package" + sugarcane	2	
	"joinmap program" + sugarcane	1,070	
Pergola	Citas en la página web	0	2
	"pergola software" + sugarcane	0	
	"pergola package" + sugarcane	2	
	"pergola program" + sugarcane	0	
Netgwas	Citas en la página web	0	2
	"netgwas software" + sugarcane	0	
	"netgwas package" + sugarcane	2	
	"netgwas program" + sugarcane	0	
OneMap	Citas en la página web	0	104
	"OneMap software" + sugarcane	83	
	"OneMap package" + sugarcane	16	
	"OneMap program" + sugarcane	5	
polymapR	Citas en la página web	0	3
	"polymapR software" + sugarcane	1	
	"polymapR package" + sugarcane	2	
	"polymapR program" + sugarcane	0	

A.4.2. Pruebas hechas en cada estrategia usando JoinMap

Tabla A.6: Experimentos con la estrategia A al variar dos parámetros para construir el mapa genético en JM, indLOD y FR. [28]

Parámetros		Métricas						
Valor indLOD	**Frecuencia de recombinación**	**Número de LGs creados**	**LG más grande lg(loci)**	**LG más pequeño lg(loci)**	**LGs con 1 loci**	**LGs con 2-19 loci**	**LGs con 20-100 loci**	**LGs con >100 loci**
9	**0.1**	1347	1(1045)	1347(1)	1059	270	16	2
9.2	**0.15**	1390	1(1023)	1390(1)	1100	273	15	2
9.4	**0.2**	1440	1(985)	1440(1)	1143	280	15	2
9.6	**0.25**	1496	1(926)	1496(1)	1202	278	15	1
9.8	**0.3**	1556	1(888)	1556(1)	1259	282	14	1
10	**0.35**	1611	1(865)	1611(1)	1318	278	14	1

Tabla A.7: Experimentos con la estrategia B al variar dos parámetros para construir el mapa genético en JM, indLOD y FR. [28]

Parámetros		Métricas						
Valor indLOD	**Frecuencia de recombinación**	**Número de LGs creados**	**LG más grande lg(loci)**	**LG más pequeño lg(loci)**	**LGs con 1 loci**	**LGs con 2-19 loci**	**LGs con 20-100 loci**	**LGs con >100 loci**
9	**0.015**	2678	1(166)	2678(1)	2396	269	11	2
9.2	**0.05**	1721	1(481)	1721(1)	1372	327	19	3
9.4	**0.1**	1435	1(593)	1435(1)	1088	322	22	3
9.6	**0.15**	1457	1(592)	1457(1)	1111	321	22	3
9.8	**0.2**	1500	1(561)	1500(1)	1158	317	22	3
10	**0.25**	1549	1(551)	1549(1)	1202	322	22	3
11	**0.3**	1766	1(456)	1766(1)	1429	314	20	3
12	**0.35**	2036	1(200)	2036(1)	1700	319	15	2
13	**0.4**	2295	1(133)	2295(1)	1946	330	18	1
14	**0.45**	2547	1(113)	2547(1)	2221	309	16	1
15	**0.5**	2846	1(101)	2846(1)	2560	272	13	1

Tabla A.8: Experimentos con la estrategia C al variar dos parámetros para construir el mapa genético en JM, indLOD y FR. [28]

Parámetros		Métricas						
Valor indLOD	**Frecuencia de recombinación**	**Número de LGs creados**	**LG más grande lg(loci)**	**LG más pequeño lg(loci)**	**LGs con 1 loci**	**LGs con 2-19 loci**	**LGs con 20-100 loci**	**LGs con >100 loci**
9	**0.015**	6829	1(119)	6829(1)	6624	194	9	2
9.2	**0.05**	5884	1(302)	5884(1)	5601	266	13	4
9.4	**0.1**	4668	1(1125)	4668(1)	4267	379	19	3
9.6	**0.15**	3924	1(2133)	3924(1)	3492	415	15	2
9.8	**0.2**	4009	1(1931)	4009(1)	3580	413	13	3
10	**0.25**	4111	1(1787)	4111(1)	3677	416	16	2
11	**0.3**	4584	1(1303)	4584(1)	4198	365	18	3
12	**0.35**	5021	1(990)	5021(1)	4665	336	17	3
13	**0.4**	5375	1(439)	5375(1)	5040	317	14	4
14	**0.45**	5634	1(381)	5634(1)	5323	293	14	4
15	**0.5**	5901	1(293)	5901(1)	5612	271	14	4

A.4.3. Convertir a formato JoinMap

Algoritmo A.20: instrucciones usando NGSEP para hacer convertir el formato `.vcf` al formato aceptado en JoinMap. [8]

```
# Autor(es): Giann Karlo Aguirre Samboní.
# Nombre del programa: 18_convertvcf_joinmap.sh
# Fecha: 21-09-2020
# Objetivo: convertir el formato .vcf al formato aceptado para JoinMap v5.0
# Salida: archivo .txt de toda la población con los loci que pasaron todos los filtros.
# Entrada: archivo .vcf resultado de la ejecución de 13_filtervcf_fourthfilter.sh.

java -jar ../ngsep_lib/NGSEPcore_4.0.1.jar VCFConverter -i lod9_rf01_lg2-18.vcf -joinMap -p1 CC_01746 -p2 CC_011940 -o lod9_rf01_lg2-18

#Ejecución de NGSEP con la instrucción VCFConverter (NGSEP v4.0.1):
# -i es un parámetro para indicar que el archivo de entrada es lod9_rf01_lg2-18.vcf.
# -joinMap es el parámetro para indicar que se requiere transformar al formato aceptado por JoinMap.
# -p1 parámetro para indicar que CC_01746 es el primer parental de la población.
# -p2 parámetro para indicar que CC_011940 es el segundo parental de la población.
# -o parámetro para indicar que lod9_rf01_lg2-18 es el nombre del archivo de salida.

# A pesar que NGSEP convierte al formato aceptado por JoinMap, se deben hacer unos ajustes a la salida del archivo:
# 1. Eliminar el encabezado que pone NGSEP.
# 2. Agregar, en cambio, este encabezado (antes del contenido del archivo):
#       name = CP_biparental (nombre de la población)
#       popt = CP   (tipo de población, p. ej. Cross pollinators)
#       nloc = 3739 (número de loci en la población)
#       nind = 95   (número de individuos en la población)

# Esta documentación está basada en la versión usada de NGSEP, así que cualquier inconsistencia por el uso de versiones, es mejor revisar la documentación oficial de NGSEP: https://github.com/NGSEP/NGSEPcore
```

A.4.4. Ejemplo archivo JoinMap

Archivo A.21: Archivo ejemplo con un muestra de datos de los marcadores de la estrategia A.

```
name = CP_biparental
popt = CP
nloc = 6
nind = 11


sc_12997_277971 <nnxnp> (nn,np) nn  np  nn  np  nn  --  --  nn  nn  nn  nn
sc_12997_291911 <nnxnp> (nn,np) nn  np  nn  np  nn  nn  np  nn  nn  --  nn
sc_12997_292151 <nnxnp> (nn,np) nn  np  nn  np  nn  nn  --  nn  nn  --  nn
sc_12997_339029 <nnxnp> (nn,np) nn  np  --  np  nn  nn  --  nn  nn  --  nn
sc_12997_473350 <nnxnp> (nn,np) nn  np  --  --  nn  np  np  --  nn  --  nn
sc_7582_12094 <hkxhk> (hh,hk,kk)  hk  hk  kk  hk  --  hk  --  hk  --  hh
```

Referencias

Artículos científicos

Awe, G., Reichert, J. y Fontanela, E. (2020). Sugarcane production in the subtropics: seasonal changes in soil properties and crop yield in no-tillage, inverting and minimum tillage. En: *Soil and Tillage Research* 196 (104447). DOI: 10.1016/j.still.2019.104447 (página 1).

Balsalobre, T., Pereira, G., Rodrigues, G., Gazaffi, R., Zatti, F. *et al.* (2017). GBS-based single dosage markers for linkage and QTL mapping allow gene mining for yield-related traits in sugarcane. En: *BMC Genomics* 18 (1), p. 72. DOI: 10.1186/s12864-016-3383-x (páginas 9, 44-46).

Bansal, V. y Boucher, C. (2019). Sequencing technologies and analyses: where have we been and where are we going? En: *Iscience* 18, pp. 37-41. DOI: 10.1016/j.isci.2019.06.035 (página 2).

Berkman, P., Bundock, P., Casu, R., Henrand, R., Rae, A. *et al.* (2014). A survey sequence comparison of Saccharum genotypes reveals allelic diversity differences. En: *Tropical Plant Biology* 7 (2), pp. 71-83. DOI: 10.1007/s12042-014-9139-3 (página 43).

Bourke, P., van-Guesst, G., Voorrips, R., Jansen, J., Kranenburg, T. *et al.* (2018). polymapR-linkage analysis and genetic map construction from F1 populations of outcrossing polyploids. En: *Bioinformatics* 34 (20), pp. 3496-3502. DOI: 10.1093/bioinformatics/bty371 (páginas 3, 28).

Bourke, P., Voorrips, R., Visser, R. y Maliepaard, C. (2018). Tools for genetic studies in experimental populations of polyploids. En: *Frontiers in Plant Science* 9, p. 513. DOI: 10.3389/fpls.2018.00513 (páginas 44, 45).

Costa, E., Anoni, C., Mancini, M., Santos, F., Marconi, T. *et al.* (2016). QTL mapping including codominant SNP markers with ploidy level information in a sugarcane progeny. En: *Euphytica* 211, pp. 1-16. DOI: 10.1007/s10681-016-1746-7 (página 9).

Cuadrado, A., Acevedo, R., Moreno, S., Jouve, N. y De La Torre, C. (2004). Genome remodelling in three modern S. officinarum × S. spontaneum sugarcane cultivars. En: *Journal of Experimental Botany* 55 (398), pp. 847-854. DOI: 10.1093/jxb/erh093 (páginas 2, 46).

D'Hont, A. (2005). Unraveling the genome structure of polyploids using FISH and GISH; examples of sugarcane and banana. En: *Cytogenetic and genome research* 109 (1-3), pp. 27-33. DOI: 10.1159/000082378 (páginas 2, 46).

D'Hont, A., Grivet, L., Feldmann, P., Glaszmann, J., Rao, S. *et al.* (1996). Characterisation of the double genome structure of modern sugarcane cultivars (Saccharum spp.) by molecular cytogenetics. En: *Molecular & General Genetics* 250 (4), pp. 405-413. DOI: 10.1007/bf02174028 (páginas 2, 46).

D'Hont, A., Ison, D., Alix, K., Roux, C. y Glaszmann, J. (1998). Determination of basic chromosome numbers in the genus Saccharum by physical mapping of ribosomal RNA genes. En: *Genome* 41 (2), pp. 221-225. DOI: 10.1139/g98-023 (página 2).

Darrier, B., Russell, J., Milner, S., Hedleand, P., P, S. *et al.* (2019). A comparison of mainstream genotyping platforms for the evaluation and use of barley genetic resources. En: *Frontiers in Plant Science* 10, p. 544. DOI: 10.3389/fpls.2019.00544 (página 43).

Daveand, J. y Blaxter, M. (2010). RADSeq: next-generation population genetics. En: *Briefing in Functional Genomics* 9 (5), pp. 416-423. DOI: 10.1093/bfgp/elq031 (página 14).

Dellaporta, S., Wood, J. y Hicks, J. (1983). A plant DNA minipreparation: Version II. En: *Plant Molecular Biology Reporter* 1 (4), pp. 19-21. DOI: 10.1007/BF02712670 (página 4).

Dohm, J., Lottaz, C., Borodina, T. y Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. En: *Nucleic Acids Research* 36 (16), e105. DOI: 10.1093/nar/gkn425 (página 21).

Duitama, J., Quintero, J., Cruz, D., Quintero, C., Hubmann, G. *et al.* (2014). An integrated framework for discovery and genotyping of genomic variants from high-throughput sequencing experiments. En: *Nucleic Acids Research* 42 (6), e44. DOI: 10.1093/nar/gkt1381 (página 5).

Elshire, R., Glaubitz, J., Sun, Q., Poland, J., Kawamoto, K. *et al.* (2011). A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. En: *PLoS ONE* 6 (5), e19379. DOI: 10.1371/journal.pone.0019379 (página 14).

Ewin, B. y Green, P. (1998). Base-calling of automated sequencer traces using phred. ii. error probabilities. En: *Genome Research* 8, pp. 186-194. DOI: 10.1101/gr.8.3.186 (página 13).

Garcia, A., Kido, E., Meza, A., Souza, H., Pinto, L. *et al.* (2006). Development of an integrated genetic map of a sugarcane (Saccharum spp.) commercial cross, based on a maximum-likelihood approach for estimation of linkage and linkage phases. En: *Theoretical and Applied Genetics* 112 (2), pp. 298-314. DOI: 10.1007/s00122-005-0129-6 (página 45).

Garsmeur, O., Droc, G., Antonise, R., Grimwood, J., Potier, B. *et al.* (2018). A mosaic monoploid reference sequence for the highly complex genome of sugarcane. En: *Nature Communications* 9 (2638). DOI: 10.1038/s41467-018-05051-5 (páginas 6, 7, 9, 41, 43-46).

Grandke, F., Ranganathan, S., van-Bers, N., de-Haan, J. y Metzler, D. (2017). PERGOLA: fast and deterministic linkage mapping of polyploids. En: *BMC Bioinformatics* 18 (12), pp. 1-9. DOI: 10.1186/s12859-016-1416-8 (páginas 3, 28).

Grivet, L. y Arruda, P. (2002). Sugarcane genomics: depicting the complex genome of an important tropical crop. En: *Current Opinion in Plant Biology* 5 (2), pp. 122-127. DOI: 10.1016/s1369-5266(02)00234-0 (página 2).

Gutierrez, F., Hoy, J., Kimbeng, C. y Baisakh, N. (2018). Identification of genomic regions controlling leaf scald resistance in sugarcane using a bi-parental mapping population and selective genotyping by sequencing. En: *Frontiers in Plant Science* 9, p. 877. DOI: 10.3389/fpls.2018.00877 (página 9).

Hoffmann, S., Otto, C., Kurtz, S., Sharma, C., Khaitovich, P. *et al.* (2009). Fast mapping of short sequences with mismatches, insertions and deletions using index structures. En: *PLOS Computational Biology* 5 (9), e1000502. DOI: 10.1371/journal.pcbi.1000502 (página 21).

Jannoo, N., Grivet, L., Chantret, N., Garsmeur, O., Glaszmann, J. *et al.* (2007). Orthologous comparison in a gene-rich region among grasses reveals stability in the sugarcane polyploid genome. En: *The Plant Journal* 50 (4), pp. 574-585. DOI: 10.1111/j.1365-313X.2007.03082.x (página 1).

Korbel, J., Urban, A., Affourtit, J., Godwin, B., Grubert, F. *et al.* (2007). Paired-end mapping reveals extensive structural variation in the human genome. En: *Science* 318 (5849), pp. 420-426. DOI: 10.1126/science.1149504 (página 10).

Lander, E. y Waterman, M. (1988). Genomic mapping by fingerprinting random clones: A mathematical analysis. En: *Genomics* 2 (3), pp. 231-239. DOI: 10.1016/0888-7543(88)90007-9 (página 15).

Langmead, B. y Salzberg, S. (2012). Fast gapped-read alignment with Bowtie 2. En: *Nature Methods* 9, pp. 357-359. DOI: 10.1038/nmeth.1923 (página 5).

Lawrence, J., Arcila, M., Corless, C., Kamel-Reid, S., Lubin, I. *et al.* (2017). Guidelines for validation of next-generation sequencing-based oncology panels: a joint consensus recommendation of the association for molecular pathology and college of american pathologists. En: *The Journal of Molecular Diagnostics* 19 (3), pp. 341-365. DOI: 10.1016/j.jmoldx.2017.01.011 (página 21).

Le Cunff, L., Garsmeur, O., Raboin, L., Pauquet, J., Telismart, H. *et al.* (2008). Diploid/polyploid syntenic shuttle mapping and haplotype-specific chromosome walking toward a rust resistance gene (Bru1) in highly polyploid sugarcane (2n approximately 12x approximately 115). En: *Genetics* 180 (1), pp. 649-660. DOI: 10.1534/genetics.108.091355 (página 2).

Li, H., Handsaker, B., Wandsoker, A., Fennell, T., Ruan, J. *et al.* (2009). The Sequence Alignment/Map format and SAMtools. En: *Bioinformatics* 25 (16), pp. 2078-2079. DOI: 10.1093/bioinformatics/btp352 (página 5).

Lieberman-Aiden, E., Van Berkum, N., Williams, L., Imakaev, M., Ragoczand, T. *et al.* (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. En: *Science* 326 (5950), pp. 289-293. DOI: 10.1126/science.1181369 (página 2).

Lowry, D., Hoban, S., Kelley, J., Lotterhos, K., Reed, L. *et al.* (2017). Breaking RAD: an evaluation of the utility of restriction site-associated DNA sequencing for genome scans of adaptation. En: *Molecular Ecology Resources* 17 (2), pp. 142-152. DOI: 10.1111/1755-0998.12635 (página 43).

Margarido, G., Souza, A. y Garcia, A. (2007). OneMap: software for genetic mapping in outcrossing species. En: *Hereditas* 144 (3), pp. 78-79. DOI: 10.1111/j.2007.0018-0661.02000.x (páginas 3, 28).

Meyer, M. y Kircher, M. (2010). Illumina sequencing library preparation for highly multiplexed target capture and sequencing. En: *Cold Spring Harbor Protocols* 2010 (6). DOI: 10.1101/pdb.prot5448 (páginas 2, 21).

Miller, M., Dunham, J., Amores, A., Cresko, W. y Johnson, E. (2007). Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. En: *Genome research* 17 (2), pp. 240-248. DOI: 10.1101/gr.5681207 (página 14).

Ming, R., Liu, S., Lin, Y., Silva, J. da, Wilson, W. *et al.* (1998). Detailed alignment of saccharum and sorghum chromosomes: comparative organization of closely related diploid and polyploid genomes. En: *Genetics* 150.4, pp. 1663-1682 (página 2).

Mohan, C. (2016). Genome editing in sugarcane: challenges ahead. En: *Frontiers in Plant Science* 7, p. 1542. DOI: 10.3389/fpls.2016.01542 (página 2).

Perea, C., De La Hoz, J., Cruz, D., Lobaton, J., Izquierdo, P. *et al.* (2016). Bioinformatic analysis of genotype by sequencing (GBS) data with NGSEP. En: *BMC Genomics* 17 (498), pp. 539-551. DOI: 10.1186/s12864-016-2827-7 (página 6).

Piperidis, G., Piperidis, N. y D'Hont, A. (2010). Molecular cytogenetic investigation of chromosome composition and transmission in sugarcane. En: *Molecular Genetics and Genomics* 284 (1), pp. 65-73. DOI: 10.1007/s00438-010-0546-3 (páginas 2, 46).

Piperidis, N. y D'Hont, A. (2020). Sugarcane genome architecture decrypted with chromosome-specific oligo probes. En: *The Plant Journa*. DOI: 10.1111/tpj.14881 (página 43).

Rhoads, A. y Au, K. (2015). PacBio sequencing and its applications. En: *Genomics Proteomics & Bioinformatics* 13 (5), pp. 278-289. DOI: 10.1016/j.gpb.2015.08.002 (página 2).

Silva, J. da, Sorrells, M., Burnquist, W. y Tanksleand, S. (1993). RFLP linkage map and genome analysis of Saccharum spontaneum. En: *Genome* 36 (4), pp. 782-791. DOI: 10.1139/g93-103 (página 44).

Sims, D., Sudberand, I., Ilott, N., Heger, A. y Ponting, C. (2014). Sequencing depth and coverage: key considerations in genomic analyses. En: *Nature Reviews Genetics* 15, pp. 121-132. DOI: 10.1038/nrg3642 (páginas 10, 43).

Souza, G., Berges, H., Bocs, S., Casu, R., D'Hont, A. *et al.* (2011). The sugarcane genome challenge: strategies for sequencing a highly complex genome. En: *Tropical Plant Biology* 4 (3), pp. 145-156. DOI: 10.1007/s12042-011-9079-0 (página 43).

Stam (1993). Construction of integrated genetic linkage maps by means of a new computer package: JoinMap. En: *The Plant Journal* 3 (5), pp. 739-744. DOI: 10.1111/j.1365-313X.1993.00739.x (páginas 3, 28).

Tello, D., Gil, J., Loaiza, C., Riascos, J., Cardozo, N. *et al.* (2019). NGSEP3: accurate variant calling across species and sequencing protocols. En: *Bioinformatics* 35 (22), pp. 4716-4723. DOI: 10.1093/bioinformatics/btz275 (página 5).

Thirugnanasambandam, P., Hoang, N. y Henrand, R. (2018). The challenge of analyzing the sugarcane genome. En: *Frontiers in Plant Science* 9, pp. 616-633. DOI: 10.3389/fpls.2018.00616 (páginas 2, 43).

Wickland, D., Battu, G., Hudson, K., Diers, B. y Hudson, M. (2017). A comparison of genotyping-by-sequencing analysis methods on low-coverage crop datasets shows advantages of a new workflow, GB-eaSy. En: *BMC Bioinformatics* 18 (1), p. 586. DOI: 10.1186/s12859-017-2000-6 (página 43).

Wu, K., Burnquist, W., Sorrells, M., Tew, T., Moore, P. *et al.* (1992). The detection and estimation of linkage in polyploids using single-dose restriction fragments. En: *Theoretical and Applied Genetics* 83 (3), pp. 294-300. DOI: 10.1007/BF00224274 (página 44).

Zhang, J., Nagai, C., Yu, Q., Pan, Y., Aandala-Silva, T. *et al.* (2012). Genome size variation in three Saccharum species. En: *Euphytica* 185 (3), pp. 511-519. DOI: 10.1007/s10681-012-0664-6 (página 2).

Zhang, J., Zhang, X., Tang, H., Zhang, Q., Hua, X. *et al.* (2018). Allele-defined genome of the autopolyploid sugarcane Saccharum spontaneum L. En: *Nature Genetics* 50, pp. 1565-1573. DOI: 10.1038/s41588-018-0237-2 (páginas 2, 3, 46).

Capítulos de libro

Boopathi, N. (2020). QTL Analysis. En: *Genetic Mapping and Marker Assisted Selection*. second. Springer Singapore. Cap. 7, pp. 253-326. DOI: 10.1007/978-981-15-2949-8 (página 3).

Donzelli, J., Bertolani, F. y de-Campos, T. (2018). Sugarcane cultivation: soil mapping, environmental effects, and new sugarcane varieties. En: Chandel, A. y Luciano, M. *Advances in Sugarcane Biorefinery: Technologies, Commercialization, Policy Issues and Paradigm Shift for Bioethanol and By-Products*. first. Elsevier Inc. Cap. 1, pp. 1-15. DOI: 10.1016/B978-0-12-804534-3.00001-X (página 1).

Gazaffi, R., Oliveira, K., Souza, A. y Garcia, A. (2014). Sugarcane: breeding methods and genetic mapping. En: Barbosa, L. *Sugarcane bioethanol - R&D for Productivity and Sustainability*. Blucher. Cap. 33, pp. 333-344. DOI: 10.5151/BlucherOA-Sugarcane-SUGARCANEBIOETHANOL_33 (página 45).

Hultén, M. y Tease, C. (2006). Genetic maps: direct meiotic analysis. En: Chichester: John Wiley & Sons, Ltd. Cap. NA, pp. 9903-9908. DOI: 10.1002/9780470015902.a0006250 (página 3).

Kapoor, M., Panwar, D. y Kaira, G. (2016). Bioprocesses for enzyme production using agro-industrial wastes: technical challenges and commercialization potential. En: Dhillon, G. y Kaur, S. *Agro-Industrial Wastes as Feedstock for Enzyme Production: Apply and Exploit the Emerging*

and Valuable Use Options of Waste Biomass. Elsevier Inc. Cap. 3, pp. 61-93. DOI: 10.1016/B978-0-12-802392-1.00003-4 (página 1).

Little, P. (2005). Genetic mapping and positional cloning. En: Chichester: John Wiley & Sons, Ltd. Cap. NA, pp. 9545-9551. DOI: 10.1038/npg.els.0005370 (página 3).

Miranda, E. de y Fonseca, M. (2020). Sugarcane: food production, energy, and environment. En: Santos, F., Matos, M. de, Eichler, P. y Rabelo, S. *Sugarcane biorefinery, technology and perspectives*. Academic Press. Cap. 4, pp. 67-88. DOI: 10.1016/b978-0-12-814236-3.00004-4 (página 1).

OECD y FAO (2018). Sugar. En: *OECD-FAO Agricultural Outlook 2018-2027*. OECD Publishing. Cap. 5, pp. 139-148. DOI: 10.1787/agr_outlook-2018-en (página 1).

Paz, M. y Shoemaker, R. (2005). Genetic and physical map correlation. En: Chichester: John Wiley & Sons, Ltd. Cap. NA, pp. 9672-9678. DOI: 10.1038/npg.els.0003937 (página 3).

Zhang, J., Zhou, M., Walsh, J., Zhu, L., Chen, Y. *et al.* (2013). Sugarcane genetics and genomics. En: Moore, P. y Botha, F. *Sugarcane: Physiology, Biochemistry, and Functional Biology*. John Wiley & Sons, Ltd. Cap. 23, pp. 623-643. DOI: 10.1002/9781118771280.ch2 (página 44).

Conferencias

D'Hont, A. y Glazsmann, J. (2001). "Sugarcane genome analysis with molecular markers: a first decade of research". En: *Proceedings International Society of Sugar Cane Technologists*. Vol. 24, pp. 556-559 (página 2).

Hoy, J., Baisakh, N., Avellaneda, M., Kimbeng, C. y Hale, A. (2016). "Detection, breeding, and selection of durable resistance to brown rust in sugarcane". En: *Proceedings of the International Society of Sugar Cane Technologists*. Vol. 29, pp. 1034-1039 (página 45).

Kilian, A., Huttner, E., Wenzl, P., Jaccoud, D., Carling, J. *et al.* (2005). "The fast and the cheap: SNP and DArT-based whole genome profiling for crop improvement". En: *Proceedings of the International Congress in the Wake of the Double Helix: From the Green Revolution to the Gene Revolution*. Ed. por P. R. Tuberosa R y G. M. Avenue Media, pp. 443-461 (página 45).

Saraswathy, N. y Ramalingam, P. (2011). "Genome mapping". En: *Concepts and Techniques in Genomics and Proteomics*. Oxford: Woodhead Publishing, pp. 77-93. DOI: 10.1533/9781908818058.77 (página 3).

Tesis doctorales

Oliveira, K. (2006). "Desenvolvimento de marcadores moleculares EST-SSRs e mapeamento funcional em cana-de-açucar (Saccharum spp.)" Tesis doctoral. Campinas, Brasil: Universidade Estadual de Campinas (página 45).

Trujillo, J. (2020). "Construcción de un genoma y una huella molecular de caña de azúcar utilizando secuenciación de alto rendimiento". Tesis doctoral. Cali, Colombia: Universidad del Valle (páginas 5, 14, 15, 41, 43).

Libros

Pierce, B. (2016). *Genetics a conceptual approach sixth edition*. Nueva York, Estados Unidos: W. H. Freeman y Company (página 45).

Reportes técnicos

Victoria, J., Viveros, C., Salazar, F., Ángel, J., Bustillo, A. *et al.* (2013). *Catálogo de Variedades de Caña de Azúcar*. Inf. téc. Cali, Colombia (página 1).

Viveros, C. (2018). *Características agronómicas y de productividad de la variedad Cenicaña Colombia (CC) 01-1940*. Inf. téc. Cali, Colombia: CENICAÑA (páginas 1, 2).

Manuales

Arango, C., Lülle, J., Teixeira, D., Iragorri, M., Belalcázar, R. *et al.* (2018). *Informe Anual 2018*. CENICAÑA. Cali, Colombia (página 1).

Borrero, V., Lülle, J., Cardona, P., Iragorri, M., Belalcázar, R. *et al.* (2019). *Informe Anual 2019*. CENICAÑA. Cali, Colombia (página 1).

Van Ooijen, J. (2018). *JoinMap 5, Software for the calculation of genetic linkage maps in experimental populations of diploid species*. Kyazma BV. Wageningen, Netherlands (páginas 8, 23, 45, 46).

Otras referencias

Andrews, S., Krueger, F., Segonds-Pichon, A., Biggins, L. y Krueger, C. (2018). *FastQC: a quality control tool for high throughput sequence data*. URL: http://www.bioinformatics.babraham.ac.uk/projects/fastqc (página 5).

Asocaña (2019). *Aspectos generales del sector agroindustrial de la caña - Informe anual 2019 - 2020*. URL: https://www.asocana.org/modules/documentos/3/363.aspx (página 1).

Behrouzi, P., Arends, D. y Wit, E. (2017). "Netgwas: an r package for network-based genome-wide association studies". arXiv preprint arXiv:1710.01236 (páginas 3, 28).

Cuschieri, A. (2018). *Gene mapping techniques*. URL: http://staff.um.edu.mt/acus1/Newgen.pdf (página 3).

Picard Toolkit (2019). Broad Institute (página 5).