UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

A dynamic epigenetic network

Máster Universitario en Bioinformática y Biología Computacional

Autor: MADARIAGA ARAMENDI, CARLOS

Tutor 1: Serra, FRANÇOIS
Institución: BSC

Tutor 2: Al-Shahrour, FÁTIMA
Institución: CNIO

Ponente: Martínez Muñoz, GONZALO
Departamento de Ingeniería Informática - UAM

FECHA: Junio 2021

Índice general

Índice de figuras

Índice de tablas

Agradecimientos

Quiero agradecer en primera instancia a Enrique Carrillo, profesor de este máster y coordinador de la asignatura de TFM, por haberme ayudado tanto en los procesos de selección del proyecto, poniéndome en contacto con el BSC y otras entidades para poder encontrar uno que se adecuase a mi experiencia e intereses, como por estar disponible en todo momento para resolver cualquier duda relacionada.

Por otro lado, agradecer inicialmente a uno de mis tutores, François Serra, por acompañarme en todo el desarrollo del proyecto y ayudarme a que esto haya sido posible, estando disponible para ello y dándome las indicaciones necesarias sobre como proceder en varias fases, así como al resto de mis tutores, Davide, Alba y Fátima, por colaborar conmigo en todo el procedimiento e ir aportando cada uno de ellos sus conocimientos y aptitudes en las diferentes fases del proyecto. De la misma manera, agradecer a Alfonso Valencia por aceptarme en su departamento del BSC para desarrollar en él mi TFM y organizar y gestionar los diferentes grupos de trabajo, aprendiendo con ello de muchas áreas diferentes en todo el proceso.

A mis compañeros de piso, Ana, Marta, Ari, Pellümb y Daniel, por estar conmigo en todo el desarrollo del trabajo, teniendo entre todos una convivencia más que estupenda, y haciéndolo todo mucho más fácil y llevadero, animándome a seguir adelante y no rendirme hasta el final, en este año tan especial y complicado en el que estaba tan lejos de las personas con las que he trabajado.

Y por último pero no menos importante, a mi madre Amalia y a mi pareja Laura, por aguantar todos los momentos difíciles y vivir todos los momentos maravillosos a mi lado, fuesen lo intensos que fuesen, y por darme fuerzas para sacar lo mejor de mí e intentar que este proyecto fuese lo mejor posible, poniendo toda la carne en el asador.

Sin todos vosotros esto no hubiera sido posible, así que una vez más, muchísimas gracias.

Resumen

Los contactos 3D de la cromatina son capaces de afectar a las enfermedades al interactuar con diferentes variantes, dentro o fuera de los genes (*enhancers*, promotores de otros genes, etc), especialmente dentro los bucles de la cromatina. El objetivo de este proyecto es integrar la información de los bucles y las variantes para una mejor caracterización de la LLC (Leucemia Linfocítica Crónica). Exploramos si los contactos de la cromatina en 3D ayudan a explicar mecánicamente el efecto de las variantes asociadas a la LLC, mediante la integración de datos 2D procedentes de estudios de asociación en el catálogo GWAS y en DISGENET, y también datos 3D, concretamente datos Hi-C (método *all-vs-all* para capturar la conformación del cromosoma), procedentes de archivos *bam* con secuencias alineadas. Convertimos estos archivos al formato adecuado para poder ejecutar dos *loop callers* diferentes: PEAKACHU y MUSTACHE. A continuación, conservamos los bucles más significativos de ambos *loop callers*, filtramos las variantes procedentes de GWAS y DISGENET e intersectamos ambos tipos de datos juntos, 2D y 3D. Al final, anotamos las regiones de cromatina en contacto con los SNPs, algunas con más de un bucle, y analizamos, el contexto genómico de las variantes, y el contexto genómico del bucle en contacto con la variante. El análisis, que tiene en cuenta la temporalidad de la progresión de la enfermedad, revela el efecto de los SNPs en los bucles de cromatina y viceversa en la LLC, y la diferenciación de células B en los genes que se ven afectados por las variantes en los contactos de los bucles de la cromatina.

Abstract

Chromatin 3D contacts are able to affect diseases outcomes by interacting with specific variants, inside or outside the genes (enhancers, promoters of other genes, etc), especially within chromatin loops. The aim of this project is to integrate loops and variant information for a better characterization of CLL (Chronic Lymphocytic Leukemia). We explore if 3D chromatin contacts help to explain mechanistically the effect of variants associated to CLL by integrating 2D data from association studies in the GWAS catalog and DISGENET, and also 3D data, specifically Hi-C data (all-vs-all method to capture chromosome conformation), stored in bam files with aligned sequences. We convert these files to the appropriate format in order to run two different loop callers: PEAKACHU and MUSTACHE. After that we keep the most significant loops from both loop callers, filter the variants coming from GWAS and DISGENET and intersect both types of data together, 2D and 3D. At the end, we annotate the chromatin regions in contact with SNPs, some having more than one loop, and we analyze, through time analysis, the genomic context of the variants, and the genomic context of the loop in contact with the variant. The analysis, which accounts for the temporality of the disease progression, reveals obtaining the effect of SNPs in chromatin loops and viceversa in the CLL, and B-cell differentiation, in the genes that are affected by the variants found in chromatin loops contacts.

Introducción

La leucemia es un tipo de cáncer que se caracteriza por iniciarse en la médula ósea, más concretamente en las células hematopoyéticas. Si una de estas células se transforma en leucémica, deja de crecer de manera natural y lo hace de forma descontrolada, normalmente dividiéndose en nuevas células mucho más rápido de lo habitual. Esta es la principal diferencia entre las células B normales y las malignas (glóbulos blancos responsables de generar anticuerpos y que presentan antígenos). [Figura 1]

Además de crecer mucho más rápido, estas células leucémicas tardan más en morir, por lo que se acumulan en la médula ósea hasta que entran en el torrente sanguíneo. Una vez que han pasado a formar parte de la sangre, pueden extenderse a otros órganos del cuerpo, impidiendo de esta forma que las células normales presentes en el organismo puedan funcionar como deberían. Lo que diferencia a este tipo de cáncer de otros que, de igual manera, atacan a la médula ósea, es que la leucemia se genera directamente en ella, mientras que los cánceres que se generan en otro lugar y posteriormente afectan a la médula ósea, no serían considerados como leucemia.

Nos centraremos principalmente en la Leucemia Linfocítica Crónica (LLC), que es el tipo de leucemia más común en adultos. En este tipo de leucemia, las células suelen crecer lentamente, por lo que muchas pueden no tener ningún síntoma durante al menos unos años; pero con el paso del tiempo, estas células terminan extendiéndose al resto del cuerpo, como los ganglios linfáticos o el hígado. [1]

Existen dos tipos principales de LLC: con y sin mutación. Cuando las células se dividen lentamente, es típico encontrar una alta tasa de mutación en ellas, y viceversa. Más concretamente, existe un gen, el IgHV, cuyo estado de mutación conduce a un resultado diferente para los pacientes [2]. En aquellos que tienen el gen IgHV mutado, la supervivencia es mejor que en aquellos que no lo tienen.

Por ello, y como han demostrado algunos estudios, la respuesta de los pacientes a la inmunoquimioterapia se ve afectada por esta mutación, provocando algunos efectos como, por ejemplo, menor tiempo de tratamiento, menor tiempo hasta el siguiente tratamiento, respuesta inferior a la quimioterapia, mayor tasa de resistencia a la quimioterapia y menor tasa de supervivencia, pero aún se desconoce el motivo real por la que esta mutación puede ayudar a predecir el desarrollo de los pacientes.

Así pues, la razón más importante para diferenciar entre la LLC mutada y la no mutada es poder aplicar los tratamientos de forma eficiente y eficaz, ajustando el momento y la cantidad de cada dosis de tratamiento en función de cada paciente y del estadio de la enfermedad [2].

Los factores de riesgo asociados a la aparición de esta enfermedad no están del todo claros, a pesar de haberse realizado numerosos estudios entorno a ella. Uno de los factores mas aceptados es tener en el historial familiar algún problema hematológico, como podría ser cualquier tipo de linfoma o directamente casos de LLC [3]. Otros parcialmente aceptados

podrían ser la franja de edad (la mayor parte de los casos se registran en pacientes mayores de 50 años) o la exposición a algunos químicos (como algunos herbicidas, como podría ser el 'Agent Orange' o el radón).

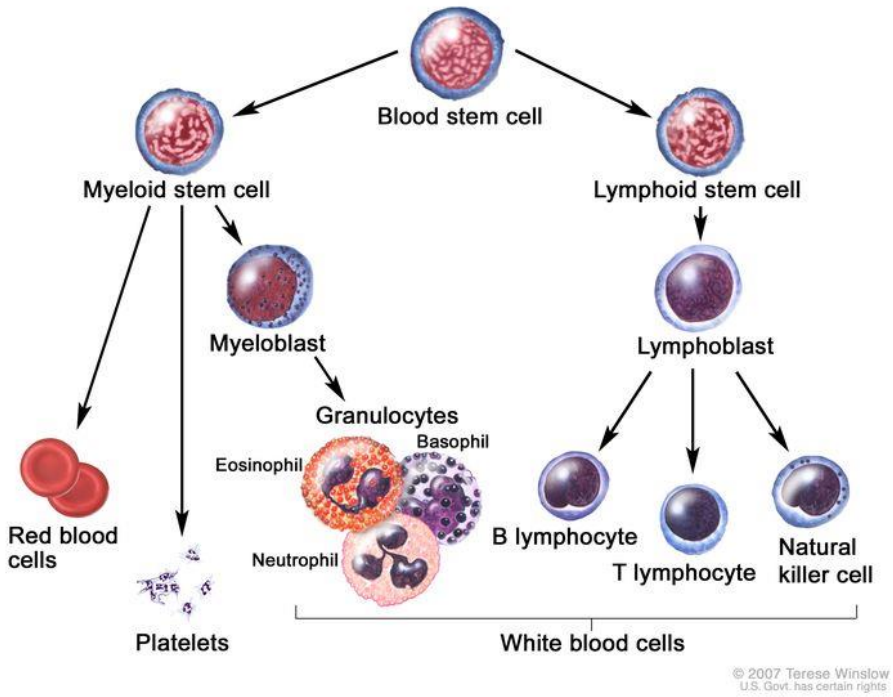


Figura 1: Desarrollo de las células sanguíneas. Una célula madre sanguínea pasa por varias fases hasta convertirse en un glóbulo rojo, una plaqueta o un glóbulo blanco. La LLC afecta a los linfocitos B. Adult Treatment Editorial Board. PDQ Chronic Lymphocytic Leukemia Treatment. Bethesda, MD: National Cancer Institute. Actualizado el 25/11/2020. Disponible en: https://www.cancer.gov/types/leukemia/patient/LLC-treatment-pdq. Consultado el 09/04/21.

También es algo más común encontrar esta enfermedad en mujeres que en hombres, y se desarrolla en mayor medida en América del Norte y Europa que en el continente asiático, lo que favorece la creencia que la aparición de esta enfermedad está más ligada a factores genéticos que medioambientales [4].

Para poder diagnosticar correctamente esta enfermedad, el primer paso es distinguirla de SLL (small lymphocytic lymphoma) a través de cómo se manifiestan, ya que LLC siempre va a afectar a células B neoplásticas. Después, es necesario comprobar que no existen otras enfermedades linfocíticas que puedan confundirse con LLC, para ello, habrá que analizar los siguientes factores:

Primero, una presencia de linfocitos en la sangre que supere 5 x 10^9/L y que se mantenga, por lo menos, durante tres meses, o bien este número sea inferior, pero exista un aumento absoluto de linfocitos B clónicos.

Segundo, observando el inmunofenotipo, ya que los niveles de inmunoglobulina superficial y de los antígenos CD20 y CD79b son especialmente altos en las células B normales, si los comparamos con los de las células existentes en LLC.

Por último, existen otras pruebas adicionales que se podrían realizar en aras de diagnosticar esta enfermedad y podrían ayudar a ello, pero que no se considerarían esenciales. Estos son, por ejemplo, citogenética molecular, cariotipado en linfocitos de la sangre, mutaciones del TP53, o como hemos mencionado previamente, observando el estado de mutación del gen IGHV. Adicionalmente, también se podría evaluar a los pacientes antes de comenzar con el tratamiento mediante pruebas médicas como un aspirado de médula y biopsia, pruebas de antiglobulina o radiografías de pecho [5].

Existen numerosas mutaciones genéticas asociadas a esta enfermedad, algunas con alta frecuencia y otras no tan comunes. Existen principalmente cuatro mutaciones comunes entre los pacientes de LLC (sobre un 80% de los pacientes poseen al menos una de ellas), denominadas supresión 13q14, supresión 11q22-23, supresión 17p12 y finalmente trisomía del cromosoma 12.

El tratamiento aplicado a los pacientes de LLC depende en gran medida de biomarcadores moleculares, como por ejemplo la perturbación del gen TP53 o bien observando si el gen IGHV está mutado o no. [6]

Por todos estos motivos, uno de los principales objetivos de este proyecto trata de encontrar los GWAS (estudios a nivel de genoma completo) que se encuentren enriquecidos en bucles, y estudiar como los SNPs (variantes de un solo nucleótido) afectan a los bucles en las diversas etapas de la diferenciación de las células B y en LLC, hasta descubrir la relación que existe entre SNPs y bucles con los genes, y poder determinar como esto influye en el tipo de leucemia y en los tratamientos aplicados en última instancia.

Estado del arte

Estudios de asociación de genoma completo: GWAS

Estos estudios (*genome-wide association studies*) se encargan de encontrar variantes en diferentes posiciones del genoma y relacionarlas con rasgos poblacionales, así como de detectar relaciones entre enfermedades comunes, como pueden ser esclerosis, diabetes, problemas psicológicos o respiratorios, y SNPs (polimorfismos de un único nucleótido). [7]

Estos tipos de mutaciones se producen cuando se sustituye una única base en el código genético, pero a pesar de sólo producirse en una base, las consecuencias pueden importantes, como por ejemplo, la eficacia al recibir ciertos tratamientos o la aparición de algún trastorno o enfermedad. Por este motivo, y al tratarse de una variación tan pequeña, los métodos encargados de su detección deben de ser muy precisos, como por ejemplo, una PCR (*polymerase chain reaction,* o reacción en cadena de la polimerasa). [8]

El origen de estos métodos data del año 2007, siendo este año en el que se publicó el primer GWAS de gran tamaño y con un buen diseño (aparecieron algunos otros anteriormente, en el 2005 y 2006, pero no fue hasta el 2007 y por los motivos mencionados, que estos estudios ganaron relevancia), para trabajar con enfermedades complejas. Éste apareció en un

artículo en la revista *Nature*, proveniente del WTCCC (*Wellcome Trust Case Control Consortium*) [75].

En, aproximadamente, cinco años a partir de ese evento, se descubrieron numerosas posiciones genómicas relacionadas con varios rasgos complejos, que previo a ese año, se desconocía por completo su relación con dichos rasgos o enfermedades [Figura 2].

Estas nuevas asociaciones descubiertas estaban relacionadas con todo tipo de enfermedades y rasgos, desde enfermedades coronarias a psicológicas, pasando por alteraciones en la estatura. Además, el número de variantes encontradas era directamente proporcional al tamaño de las muestras tomadas, ya que a mayor fuese este tamaño, mayor sería el número de variantes encontradas asociadas con rasgos y enfermedades.

Aunque normalmente las variantes encontradas no explicaban una proporción elevada de variación genética, a partir del comienzo de la era GWAS, y especialmente para ciertas enfermedades y rasgos, esta proporción creció enormemente. [7]

A partir del año 2008, además de la creciente aparición de nuevas asociaciones, se creó el 'GWAS Catalog' (ebi.ac.uk/gwas), en colaboración entre el NHGRI (*National Human Genome Research Institute*) y el EMBL – EBI (*European Bioinformatics Institute*), en el cual se recogía la información publicada sobre los estudios GWAS y se ponía a disposición de los usuarios, habiendo sido manualmente curada previamente. En el 2015, esta base de datos se rediseñó y fue movida a los servidores del EMBL – EBI, proporcionando así a partir de entonces una nueva interfaz gráfica, y otras mejoras, como la posibilidad de buscar ontologías o soporte para estudios de secuenciación. [9]

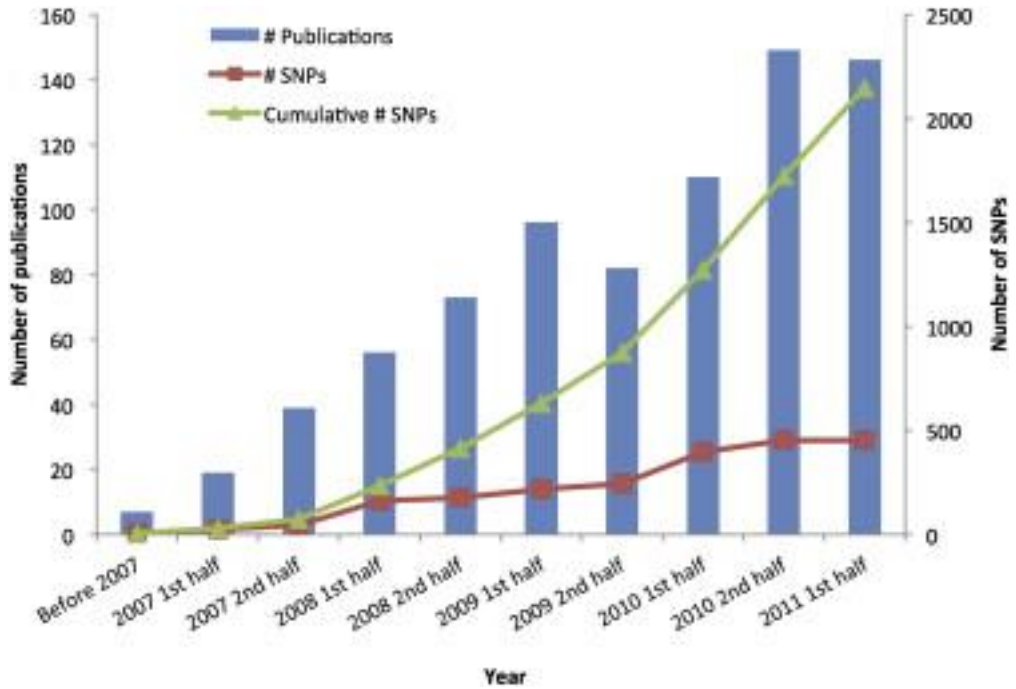


Figura 2: Descubrimientos GWAS entre el año de aparición (2007) y los cinco primeros años posteriores. Publicaciones y SNPs (puntuales y acumulados). Adaptado de [7].

En los últimos años, este catálogo ha sido ampliado en buena medida, conteniendo, a fecha de 25 de Marzo de 2021, 4691 publicaciones y 251401 asociaciones, como se puede comprobar directamente en su página. También ha sido mejorado incluyendo nuevas características, como por ejemplo dando acceso a los usuarios a estadísticas generales y aceptando contribuciones de estas estadísticas para apoyar artículos seleccionados del catálogo, siendo estos ficheros accesibles a través de una nueva página y enlazándolos con los metadatos y la información curada relacionada con éstos [10]. También se desarrolló una nueva API para el catálogo, con el fin de facilitar el acceso a la información.

Para este proyecto, una de las fuentes principales de obtención de datos sobre variantes ha sido el catálogo de GWAS, en el que se han realizado varios análisis para escoger las variantes mas relevantes para el experimento, que veremos más adelante.

Por otra parte, otra de las bases de datos utilizadas para obtener datos sobre variantes genéticas ha sido DisGeNET, de la que hablaremos a continuación.

Base de datos DisGeNET

Gracias a métodos como GWAS, cada vez se identifican más variantes y más asociaciones de éstas con trastornos y enfermedades, pero interpretar estos resultados provenientes de dichas asociaciones continúa siendo complicado, al requerir mucho tiempo y recursos para analizar todas las fuentes existentes que nos proporcionan estos datos. Para resolver este problema, es necesaria la automatización de estos procesos de interpretación a través de recursos y herramientas bioinformáticas, y en este caso, un ejemplo de ello es DisGeNET.

Esta herramienta, inicialmente, cuando fue presentada, en 2010, formaba parte de la plataforma *Cytoscape* en forma de plugin [11] y que con el paso de los años hasta la actualidad, se ha consolidado como una base de datos que contiene gran cantidad de asociaciones de tipo variante-enfermedad y de genes, en varios formatos y que contiene numerosas herramientas de exploración que ayudan a la interpretación de las diversas asociaciones presentes en ella.

Se estructura en dos tipos principales de asociaciones; la primera de ellas, denominada GDA, contiene las asociaciones gen-enfermedad, y la segunda, las de tipo variante-enfermedad, o VDA (que son las que nosotros utilizamos para el proyecto). Además, el origen de estas asociaciones se indica de tres formas diferentes; según de qué base de datos se obtuvieron originalmente, según el número de artículos en los que aparece la asociación o con una pieza del artículo que evidencia la asociación. Además, recientemente se ha agregado otro tipo de asociación a la base de datos: las relaciones enfermedad-enfermedad (DDAs, *disease-disease associations*), para poder explorar similitudes entre enfermedades observando genes y variantes compartidas entre ellas.

Una de las características más relevantes de DisGeNET es la utilización de técnicas de *TextMining* para obtener variantes de ambos tipos (GDA y VDA), con las que, por ejemplo, se han obtenido la mayor parte de las GDAs de la base de datos (entorno al 60% de estas).

Mediante estas técnicas también se comprueba que las asociaciones que aparecen en la base de datos están respaldadas por referencias bibliográficas, que se indican al usuario a través de la pieza del artículo que mencionábamos previamente.

Las variantes presentes en esta base de datos se identifican a través del identificador dbSNP [12] y se anotan con los alelos y coordenadas genómicas provenientes de la misma base de datos.

Para clasificar las variantes, DisGeNET proporciona un *score* desarrollado directamente por los creadores, que se basa en el conocimiento general para evaluar la variante, dando una mayor puntuación a aquellas que aparecen en varias bases de datos y han sido tratadas por más expertos. La fórmula, en el caso de las VDA, que son las asociaciones que nos interesan para el experimento, se computa dando un mayor peso al número de fuentes curadas que justifican dicha variante y al número de publicaciones en las que ésta aparece *(Fórmula de cáclulo del VDAScore)* [13].

La base de datos dispone además de varios métodos diferentes de acceso a la información, como a través de su API o utilizando directamente la interfaz web. Todos los datos presentes en la página siguen los principios FAIR (*Findable, Accesible, Interoperable and Reusable*), al igual que los presentes en el catálogo GWAS, lo que nos ha llevado a utilizar estas dos bases de datos para la intersección de las variantes asociadas a LLC y otras enfermedades relacionadas.

El mayor problema existente con las asociaciones variante-enfermedad obtenidas, principalmente de los GWAS, pero también de DisGeNET, es que, como se puede observar en la gráfica [Figura 3], los estudios realizados por año para entender los riesgos de enfermedades identificados por GWAS son mucho menores a los descubrimientos de nuevas asociaciones. A pesar de existir un gran número de estas asociaciones identificadas en los últimos años, pocas de ellas han sido investigadas a fondo para encontrar cuales son causales, que funciones moleculares desempeñan, a qué genes afectan o su implicación en la regulación de la expresión genética y sus consecuencias. [14].

Interacciones cromosómicas de largo alcance

La investigación de los SNPs no codificantes y como éstos afectan al riesgo de padecer una enfermedad se encuentra en continua expansión, asociando genes diana con los SNPs obtenidos mediante GWAS para numerosas enfermedades. Pero, debido a las limitaciones de las tecnologías genómicas y epigenómicas, resulta complicado distinguir qué genes ven su expresión alterada derivada de cambios en la expresión de los genes que les regulan. Por ejemplo, un cambio en la expresión de un factor de transcripción hace que los genes a los que éste regula vean afectada su expresión. Asimismo, identificar qué genes varían su expresión de forma indirecta por cambios de expresión en targets directos a estos. Por ejemplo, alteraciones en la expresión de una kinasa pueden provocar cambios de expresión en varios componentes de las vías de señalización críticas derivadas de ésta.

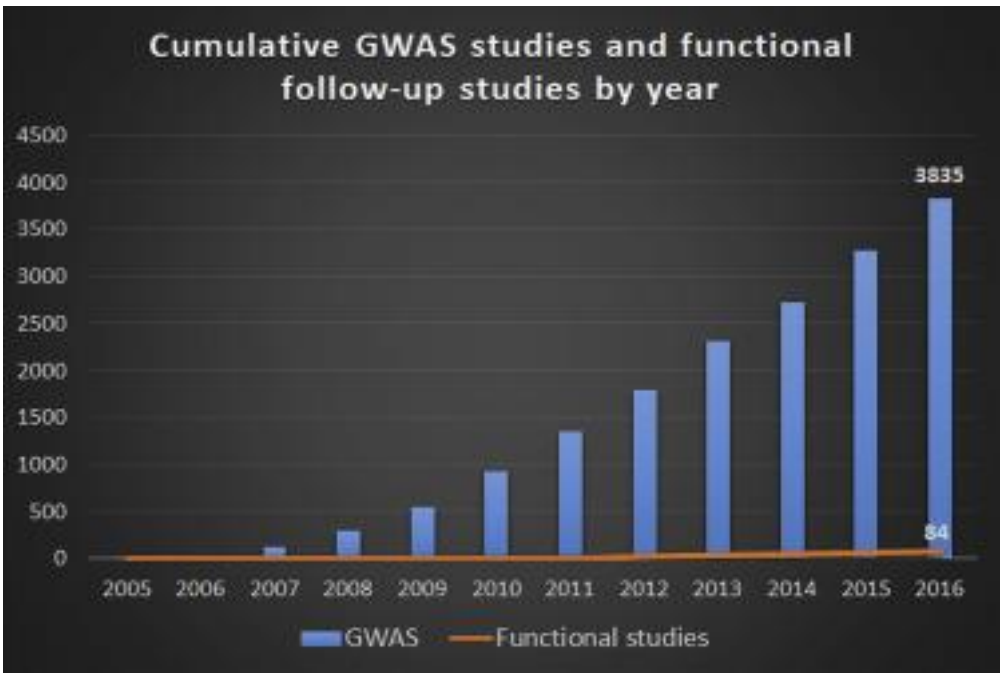


Figura 3: Comparación entre los estudios acumulados de tipo GWAS y los estudios funcionales de seguimiento realizados por año. Adaptado de [14]

Los elementos reguladores se pueden posicionar tanto cerca como lejos de los genes que regulan, y habitualmente se sitúan bastante lejos, incluso en otros cromosomas. Muchas de estas regulaciones suceden en cadena, siguiendo una combinación de proteínas *enhancer-binding*, o bien mediante bloqueo o represión de la expresión de algunos genes desde largas distancias. En estas vías de regulación, es donde una mutación puede afectar a todo el proceso, pudiendo afectar incluso a varios genes.

Para poder distinguir entre genes afectados directamente e indirectamente por un *enhancer* asociado a una enfermedad, una de las opciones es realizar ensayos de interacciones físicas entre cromosomas, utilizando por ejemplo los métodos de *Chromosome Conformation Captur*e (3C), en los que las regiones genómicas distantes se acercan mediante complejos proteínicos y posteriormente pueden ser ligadas químicamente, para después analizar el resultado.

Esto ha permitido a los investigadores estudiar las interacciones a larga distancia entre secuencias cromosómicas, que pueden suceder independientemente o en combinaciones. Es aquí donde comienza el estudio de los bucles de la cromatina, los cuales suceden cuando aparecen tramos de la secuencia genómica, que se pueden encontrar tanto en un mismo cromosoma como en cromosomas diferentes pero más cercanos físicamente entre sí que a las secuencias intermedias.

También existen otras interacciones que cumplen las mismas funciones que los bucles, pero no se consideran como tal, al darse éstas entre elementos que se encuentran en cromosomas diferentes.

Existen diversas técnicas para detectar la formación de bucles; la más utilizada es la que hemos descrito previamente: 3C. Estos ensayos necesitan numerosos controles debido a que los resultados están expuestos a muchas influencias externas, además de estar limitados a una región del genoma preseleccionada, obviando las interacciones que quedan fuera de ésta. Por ello, se han desarrollado versiones a mayor escala de ellos, como 4C o 5C, que utilizan *microarrays* o secuenciación de alto rendimiento, para facilitar el estudio de un mayor número de interacciones a lo largo del genoma.

Otro método para estudiar la formación de bucles es mediante ensayos con la inmunoprecipitación de la cromatina (ChIP), comprobando por ejemplo si una proteína de unión a un *enhancer* es detectada por ChIP en un promotor distante que carece de un espacio para unirse a este factor, puede indicar que el *enhancer* alcanza el promotor a través de un bucle [Figura 4]. Sin embargo, el *enhancer* también puede unirse al promotor a través de una proteína intermedia, por lo que en este caso sería necesario realizar alguna comprobación o experimento adicional.

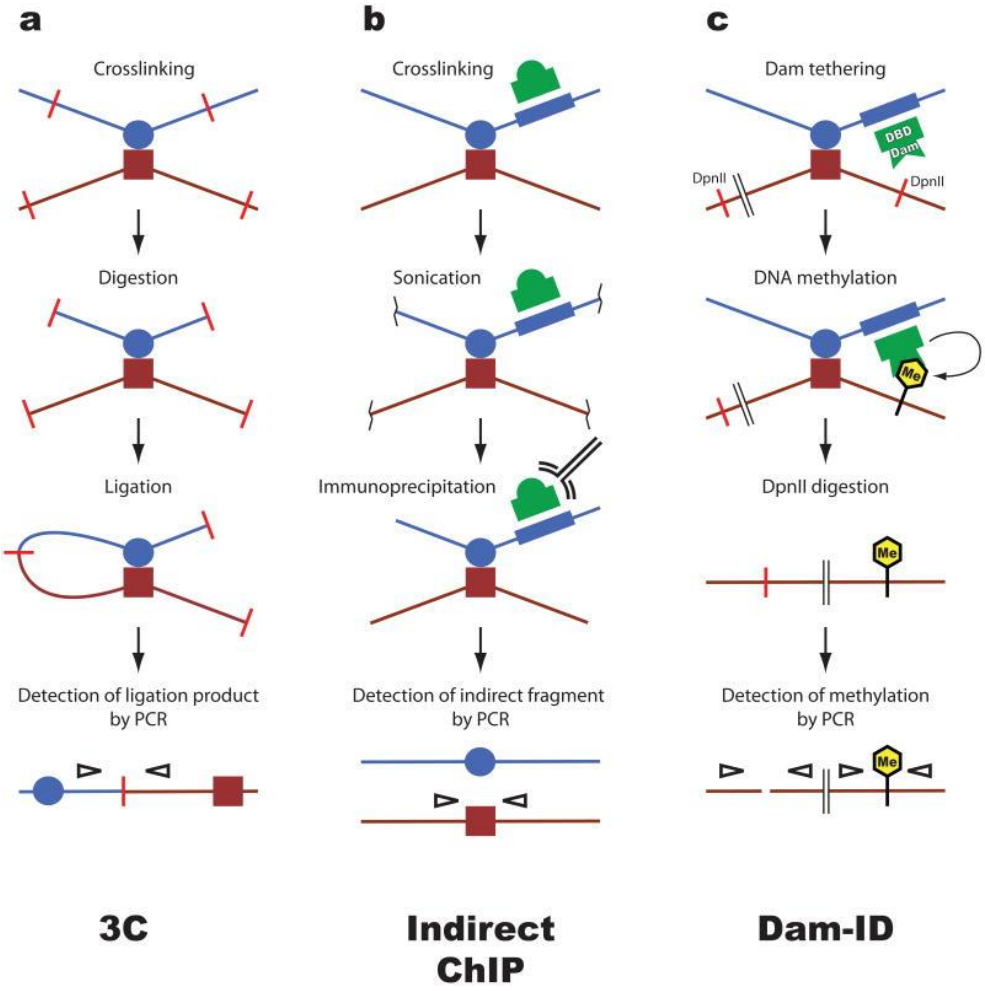


Figura 4: Métodos para detectar bucles en la cromatina. A) Chromosome conformation capture (3C). B) Inmunoprecipitación indirecta de la cromatina (Indirect ChIP). C) Indentificación de la adenina metiltransferasa del ADN (Dam-ID). Adaptado de [18]

Algunos ejemplos de bucles formados en mamíferos podrían ser las interacciones entre un potenciador distal llamado LCR (*locus control región*) y los promotores activos de la β-globina

[16] o en las células B entre los loci IgH e Igκ [17]. La organización de la cromatina en bucles en las células eucariotas es bastante habitual y pueden cumplir varias funciones genómicas como activación o represión de la transcripción o la recombinación de ADN.

A pesar de poder identificar las proteínas capaces de mediar en los bucles de la cromatina, no termina de estar claro cómo las secuencias se encuentran unas a otras para formar interacciones específicas, Algunas posibilidades estudiadas indican que podría ser mediante interacciones proteína-proteína, que estabilizarían colisiones aleatorias formando así los bucles. Otra posible forma sería mediante el desplazamiento de las proteínas ancladas por la cromatina hasta un enhancer, haciendo llegar el ADN asociado hasta los factores de unión del promotor y formando una interacción estable. [18]

Método Hi-C

En el año 2010 surgió un nuevo método para poder estudiar los pliegues y bucles de la cromatina a partir de 3C y sus sucesores, ya que éstos solo eran capaces de analizar una selección de loci objetivos, lo que imposibilitaba el estudio a nivel del genoma completo. Este método fue desarrollado por Lieberman-Aiden, van Berkum y otros autores [19] y se denominó Hi-C, que permite el estudio e identificación de interacciones a un rango mucho más amplio y sin ningún *bias*, y combina la ligación de proximidad con secuenciación masiva en paralelo. La información obtenida permite estudiar la arquitectura genómica en varias escalas, como los territorios cromosómicos, al explorar las regiones cromosómicas de interés enfrentando las interacciones “all vs all” (en el caso de 3C o 4C por ejemplo, estos enfrentamientos son “one-vs-some” o “one-vs-all” respectivamente).

El funcionamiento de este método [Figura 5] comienza etiquetando todos los fragmentos genómicos (con nucleótidos biotinilados) antes de la ligación, dejando así marcadas las uniones que posteriormente se purificarán mediante fragmentos magnéticos cubiertos por estreptavidina.

Realizando secuenciación directa de los datos obtenidos con Hi-C podemos realizar análisis con mucha potencia estadística de la organización genómica utilizando la información obtenida con este método sobre las interacciones de la cromatina, que pueden ayudar a revelar contactos específicos de largo alcance entre genes y elementos reguladores, además de la estructura general del genoma o algunas propiedades biofísicas de la cromatina.

Si además se combina la información obtenida mediante Hi-C con otros conjuntos de datos, como por ejemplo GWAS, es posible situar conjuntos de loci en contextos tridimensionales [20], lo que permite descubrir nuevas funciones de la configuración de la cromatina en regulación genómica y estabilidad, ya sea en células normales o en estados patológicos de éstas, que es una de las bases de este proyecto.

Para utilizar los métodos Hi-C orientados al estudio de la arquitectura cromosómica, se deben seguir los siguientes pasos [20]:

1) Cultivo celular y reticulación de la cromatina
2) Lisis celular y digestión de la cromatina
3) Marcar con biotina de los extremos del ADN y unir los extremos
4) Purificación del ADN
5) Control de calidad de las bibliotecas Hi-C
6) Eliminación de la biotina de los extremos no ligados
7) Fragmentación del ADN y fraccionar por tamaños
8) Reparación de los extremos y de las colas "A"
9) Extracción de estreptavidina de productos de unión Hi-C biotinilados
10) Unir adaptadores por pares y amplificar bibliotecas
11) Control de calidad final y cuantificar las bibliotecas

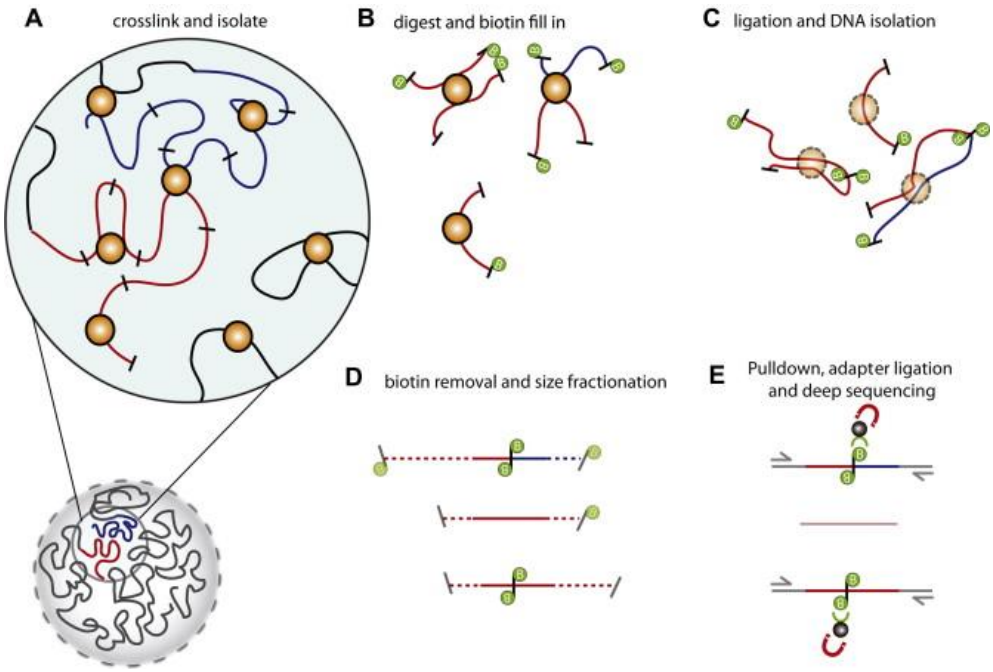


Figura 5: Funcionamiento de la tecnología Hi-C: A) reticular y aislar proteínas o complejos de ADN para detectar interacciones cromáticas, B) digerir con una encima de restricción y marcar los extremos con biotina, C) unir y aislar ADN para formar moléculas de ADN quiméricas, D) eliminar la biotina y fraccionar para reducir el tamaño global , E) desplegar con estreptavidina, unir adaptadores y secuenciación profunda masiva y en paralelo para cuantificar las interacciones de la cromatina. Adaptado de [20].

Secuenciación de nueva generación: NGS

Los datos Hi-C suelen ser obtenidos utilizando secuenciación de alta generación (NGS: *Next Generation Sequencing)*, que se trata de un tipo de tecnología de secuenciación más moderna que la de Sanger [21], aunque comparten muchos parecidos, como utilizar amplificación por PCR para preparar el ADN que se va a secuenciar.

Pero las principales diferencias que existen entre la clásica secuenciación de Sanger y la actual NGS son que esta última modalidad se aplica por moléculas de ADN individuales (Sanger las examina todas juntas en reacción), y que se generan datos sobre la secuencia en

tiempo real, ya que se va creando una cadena de ADN donde se va incorporando cada nucleótido.

De esta forma, NGS puede secuenciar el genoma humano en un solo día, o unos pocos, mientras que utilizando la secuenciación de Sanger se empleó más de una década en conseguirlo [22].

Este método de secuenciación se organiza en tres pasos principales: preparación de las bibliotecas de secuenciación, secuenciación utilizando síntesis del ADN y, por último, análisis de los datos resultantes.

El primero de todos, al igual que sucedía con el método de Sanger, implica la selección de las regiones del ADN que queremos secuenciar y su posterior amplificación, sin límite ni de complejidad ni de tamaño a la hora de elegir las regiones, pudiendo ser desde un exón de un gen hasta todos los exones de varios genes, a diferencia de al utilizar el método de Sanger, en el que el máximo tamaño de la región seleccionada se situaba en unas 500-600 bases.

El siguiente paso consta de la propia secuenciación, para lo que se utilizan las bibliotecas de fragmentos de ADN creadas en el paso anterior, se separan físicamente las moléculas de ADN y se amplifican mediante PCR. Existen dos formas principales de realizar esta separación y posterior amplificación: el primero de ellos utiliza microperlas recubiertas del oligonucleótido complementario a la secuencia, a las que posteriormente se les aplican varios procesos para después amplificarlas individualmente mediante PCR, obteniendo finalmente varias copias idénticas de la secuencia. El segundo método divide la secuencia de ADN en pequeñas 'islas' que después se secuencian utilizando pirosecuenciación [23] y ATP (adenosín trifosfato).

Tras estos pasos, se obtienen muchísimas lecturas de ADN, del orden de cientos de miles o millones, del fragmento secuenciado, y el número de veces que una secuencia se detecta en 'islas' o en las microperlas, según el método utilizado, se denomina *read depth* o profundidad de lectura, que no siempre es necesario que sea extremadamente grande. Resulta complicado y costoso analizar los datos obtenidos mediante secuenciación NGS ya que, por un lado, primero hay que alinear el genoma objetivo con las lecturas obtenidas, y después, identificar y anotar variaciones en los pares de bases tras esta alineación, lo que se trata de un proceso computacionalmente intensivo y pesado, (especialmente la alineación, ya que la anotación es mucho mas directa); para solventar este problema, existen bases de datos en las que se encuentran anotadas las variaciones encontradas en numerosos estudios [24].

Hace ya más de 15 años que los NGS fueron desarrollados, también con el nombre de métodos de secuenciación masivos en paralelo [25], existen numerosas plataformas de secuenciación NGS, como la secuenciación por ligadura, por síntesis (CRT y SNA) o mediante lecturas largas de una sola molécula en tiempo real, y actualmente forman parte de la rutina de investigación biológica y cuentan con numerosas aplicaciones en diversos campos, desde la biomedicina hasta la neurociencia, aunque el problema de la velocidad de generación de resultados aún está patente [26].

Loop Callers

Por tanto, gracias a estos métodos de secuenciación, es posible obtener datos de tipo Hi-C sobre los bucles e interacciones de largo alcance de la cromatina, y a partir de ellos estudiar cómo estas interacciones influyen en la regulación y expresión genética.

Aunque ya es conocido el efecto de la presencia o ausencia de bucles en el desarrollo de ciertas enfermedades, aún es necesario explorar cómo afectan exactamente estas diferencias, incluyendo las variaciones en las posiciones en las que estos bucles aparecen, ya que pueden cambiar la propensión a que se produzcan las interacciones, lo que se asocia a su vez a grandes cambios en los efectos funcionales derivados de estas interacciones [27].

Por este motivo, y sumado al creciente tamaño y complejidad de las fuentes de datos Hi-C existentes, es necesario realizar múltiples pasos para analizar estos datos [28], que van desde el preprocesado de la información obtenida, hasta la extracción de conclusiones biológicas viables, incluyendo la visualización de los datos obtenidos.

Este proceso, una vez obtenidos los datos Hi-C necesarios para realizar el experimento, se lleva a cabo utilizando unas herramientas denominadas *loop callers*, cuya función principal es analizar los ficheros contenedores del genoma de una línea celular, y encontrar en ellos, utilizando estrategias como por ejemplo aprendizaje supervisado, las interacciones que suceden en la cromatina, y mostrarlas de forma comprensible, indicando en qué cromosoma se producen y entre que posiciones de éste se encuentra el comienzo y final del bucle.

Su principal forma de actuación, es detectar e identificar estas interacciones utilizando mapas de probabilidad de contacto a nivel de genoma completo.

Por un lado, existen regiones en la cromatina con una elevada interacción interna, pero cuya interacción con las regiones de otros dominios es mucho más limitada; Estas regiones se llaman dominios topológicamente asociados (TADs). Por otro lado, en las matrices o mapas de interacción se recogen las interacciones que se encuentran en la cromatina, que se pueden generar utilizando las grandes cantidades de pares de lecturas que se obtienen mediante Hi-C [29].

No existe un método consensuado de clasificar estas herramientas encargadas de la detección de bucles en la cromatina, pero podríamos dividirlas en dos grupos:

En el primer grupo se sitúan los métodos que ajustan de forma global modelos estadísticos o probabilísticos a los mapas o matrices de probabilidad de contacto, y que además asignan p-valores a cada entrada del mapa comparando el recuento de valores observados respecto a los valores esperados, calculados a partir del modelo ya ajustado.

Algunas de las herramientas existentes que se situarían en este grupo, serían por ejemplo Fit-Hi-C [30] o HiC-DC [31].

En el segundo grupo, es donde encontraríamos los métodos basados en el enriquecimiento local, a las que nos referíamos previamente como *loop callers*, y que funcionan identificando

en el mapa de contactos los picos 2D que son tanto máximos locales con respecto a sus vecinos, como significativamente más altos que el conjunto global.

En este grupo podríamos situar algunos *loop callers* como HiCCUPS (*Hi-C Computational Unbiased Peak Search*) [32], SIP (*Significant Interaction Peak caller*) [33], o Mustache [34].

La mayoría de estos métodos se basan en la búsqueda de puntos estadísticamente enriquecidos en mapas a nivel de genoma completo, pero otros, como es el caso de Peakachu [35], utilizan aprendizaje automático, en este caso en forma de *random forest* [38], para detectar los bucles en la cromatina, definiendo conjuntos de datos positivos (listas de interacciones de tipos de datos ortogonales provenientes de experimentos biológicamente enriquecidos, como ChIA-PET [36] o Capture Hi-C [37], o bien imágenes de alto rendimiento) y datos negativos (interacciones con distancias genómicas iguales o mayores que las del conjunto positivo) [Figura 7].

Normalmente no es suficiente con utilizar únicamente una herramienta de detección de bucles, ya que al funcionar cada una de ellas siguiendo un método diferente, el número, tamaño y posición de los bucles obtenidos al ejecutar cada una puede llegar a ser muy diferente [Figura 6] (aunque con los *loop callers* modernos, estas diferencias son cada vez menores). La clave para realizar un análisis exhaustivo, es utilizar varias de estas herramientas y buscar un consenso con los bucles detectados por cada una de ellas, ya sea o bien con la unión o bien con la intersección.

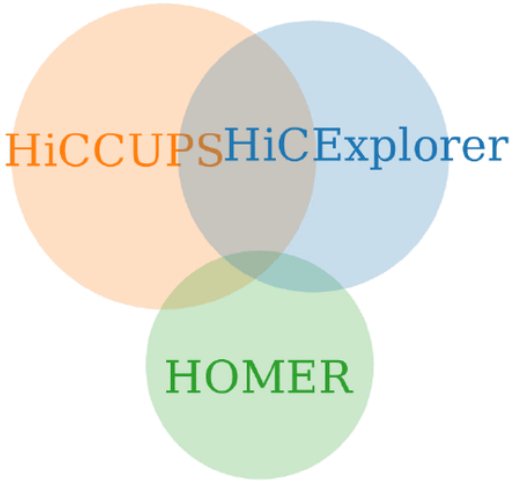


Figura 6: Ejemplo de bucles detectados por diferentes herramientas y la intersección entre éstas. Adaptado de [39]

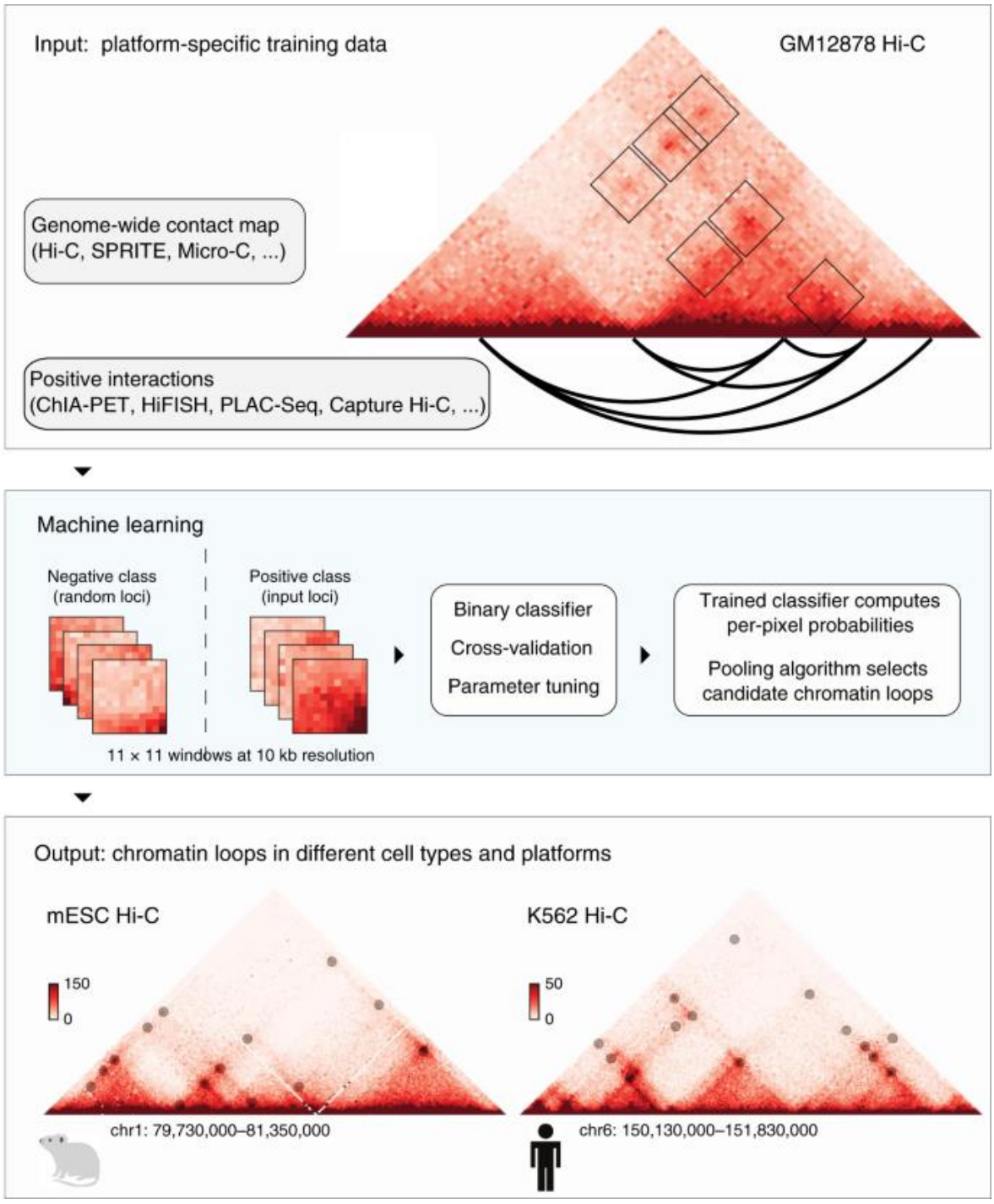


Figura 7: Funcionamiento de la herramienta Peakachu para detectar bucles en la cromatina, partiendo de una matriz de contacto proveniente de, por ejemplo, Hi-C, utilizando un modelo de aprendizaje supervisado tipo random forest para finalmente obtener estos bucles en diferentes líneas celulares y plataformas. Adaptado de [35].

Desarrollo y métodos

A lo largo del proyecto, se siguieron diversas fases tanto por el lado de los estudios de asociación y variantes, como por el de los bucles de la cromatina y las herramientas encargadas de su análisis y detección (*loops callers*) [Figura 8].

Inicialmente se realizó una exploración de los datos disponibles, así como del software que planeábamos utilizar en el desarrollo del experimento. Por un lado, se planeaba utilizar varias bases de datos para para buscar y obtener mutaciones provenientes de estudios GWAS (Genome-wide Association Studies) relacionados con la Leucemia y con otras enfermedades humanas (para utilizarlas como control).

Por otro, se realizó un análisis de las herramientas bioinformáticas existentes encargadas de detectar bucles en la cromatina utilizando datos de tipo Hi-C, para extraer contactos 3D significativos de varios tipos, como promotor-enhancer, y seleccionar las que mas se ajustaban a los requisitos y objetivos del proyecto.

A continuación, el siguiente paso fue diseñar el flujo de procesos necesarios para procesar los datos Hi-C, utilizando para ello 6 líneas celulares diferentes: CLL (*Chronic Lymphocytic Leukemia*), MCL (*Mantle Cell Lymphoma*), GCBC (*Germinal Center B-Cell*), MBC (*Memory B-Cell*), NBC (*Naive B-Cell*) y PBC (*Plasma B-Cell*), con un total de 32 muestras.

Después, utilizando estas líneas celulares, se desarrolló un algoritmo que utilizó los *loop callers* seleccionados (Mustache y Peakachu) para detectar el enriquecimiento en la carga mutacional en los bucles de nuestra base de datos, contenida en los servidores del BSC, y obtenida gracias al proyecto BLUEPRINT. Con esta información, se construyó una base de datos de GWAS y bucles, ambas con controles biológicos y aleatorios.

Finalmente, y una vez obtenidos todos los datos necesarios, se procedió a realizar los análisis relevantes para el estudio, definiendo un marco estadístico para relacionar los GWAS con los bucles de la cromatina y utilizando una dimensión temporal para el análisis, asegurando en el proceso consistencia estadística.

Tras ello, dentro del marco definido, se buscaron las asociaciones y se llevó a cabo un enriquecimiento funcional para identificar mecanismos biológicos en cuanto a los genes implicados, o bien en cuanto a las modificaciones en los puntos de unión del ADN de factores de transcripción determinados.

Por último, con los datos recogidos y los análisis realizados, se procedió a escribir este documento recogiendo todo el procedimiento realizado.

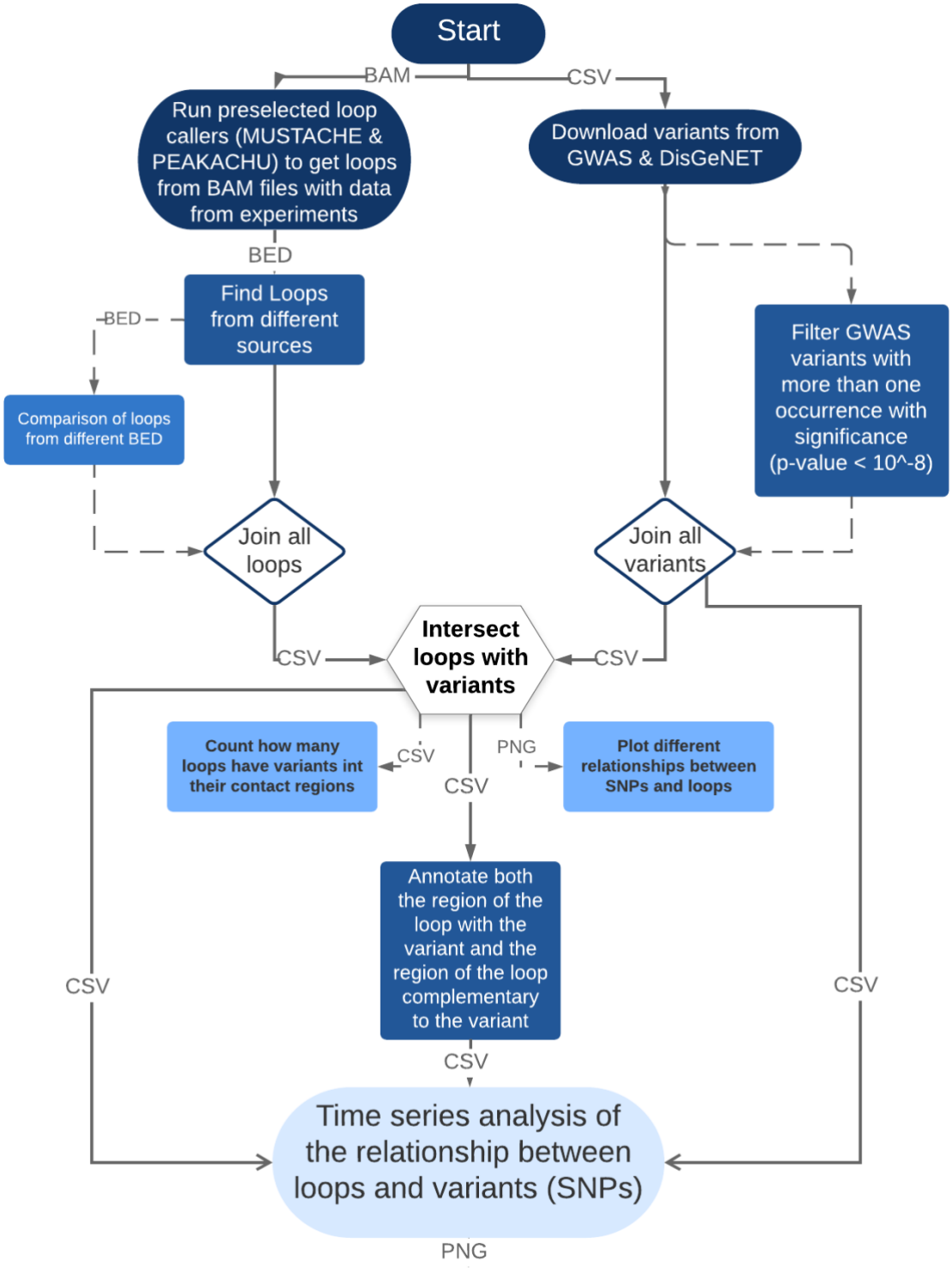


Figura 8: Flujo de trabajo de todo el proyecto, dividido inicialmente en dos vías: la parte de los bucles, con la selección inicial de las herramientas para detectar los bucles en la cromatina (*loop callers*), el análisis de los resultados y la unión de todos los bucles obtenidos (parte izquierda), y la parte de las variantes, en la que inicialmente se recogen estudios procedentes de GWAS y DisGeNET, se filtran las variantes según su p-valor, y se juntan unas con otras. Por último, se crea la intersección de bucles y variantes, y se realizan los análisis necesarios. En el flujo se indica también el tipo de fichero utilizado en cada caso, tanto el formato de entrada como el del fichero generado tras cada acción.

Variantes

Para comenzar con el experimento, partimos del trabajo realizado por parte de Paula Balcells, estudiante de máster de bioinformática en la UPF, hace dos años en el BSC, que trataba de explicar los resultados de GWAS en LLC utilizando datos Hi-C en diferentes etapas del desarrollo de las células B, y que nos servía para trazar un recorrido a lo largo de las diferentes fases del proyecto, y también para tener un punto de partida claro y definido: Obtener, por un lado, las variantes relacionadas con LLC de distintos orígenes, y por otro, detectar los bucles en la cromatina utilizando varias herramientas encargadas de esta tarea, partiendo de datos genómicos previamente obtenidos.

Obtención de las variantes

El primer paso del proyecto comenzaba analizando y obteniendo los estudios de asociación de variante-enfermedad de varias bases de datos.

Inicialmente, se consideraron tres recursos de los que obtendríamos estas variantes: GWAS Catalog [7], DisGeNET [11] y ClinVar [40]. Del primero de ellos, se podían obtener datos como por ejemplo, la variante y el alelo de riesgo, el p-valor, el gen mapeado o los rasgos reportados, así como la ubicación de la variante.

De DisGeNET se podían obtener datos similares a los anteriores, como el gen, la posición y el cromosoma en el que se encuentra la variante, el tipo de variante o el historial de referencias, así como un valor propio de la base de datos (VDAScore) indicando el respaldo que tiene cada variante basado en el número de fuentes y publicaciones en las que aparece la variante.

Sin embargo, en el caso de ClinVar, los datos que nos proporcionaba la base de datos eran bastante limitados, ya que únicamente se indica información relacionada con las condiciones, el significado clínico o el gen en el que aparece, pero al no disponer de un valor de confianza, como el p-valor del GWAS Catalog o el VDAScore de DisGeNET, finalmente decidimos no utilizar los datos provenientes de esta base de datos para realizar la unión con el resto de variantes.

Tras tomar esta decisión, comenzamos utilizando GWAS y DisGeNET para buscar únicamente el término 'Lymphoma' y por otro el término 'CLL', obtuvimos los resultados de ambas bases de datos por separado, y nos encargamos de realizar la intersección de los datos obtenidos utilizando scripts del lenguaje de programación R, en aras de comprobar la existencia de asociaciones entre ambos términos.

Nos quedamos con las variantes obtenidas, y después comprobamos en el catálogo de GWAS las asociaciones de variantes de los diferentes linfomas existentes, ya que muchos de ellos correspondían a rasgos muy variados, y debíamos que quedarnos únicamente con aquellos cuyos fenotipos coincidieran o, al menos, se relacionasen con LLC.

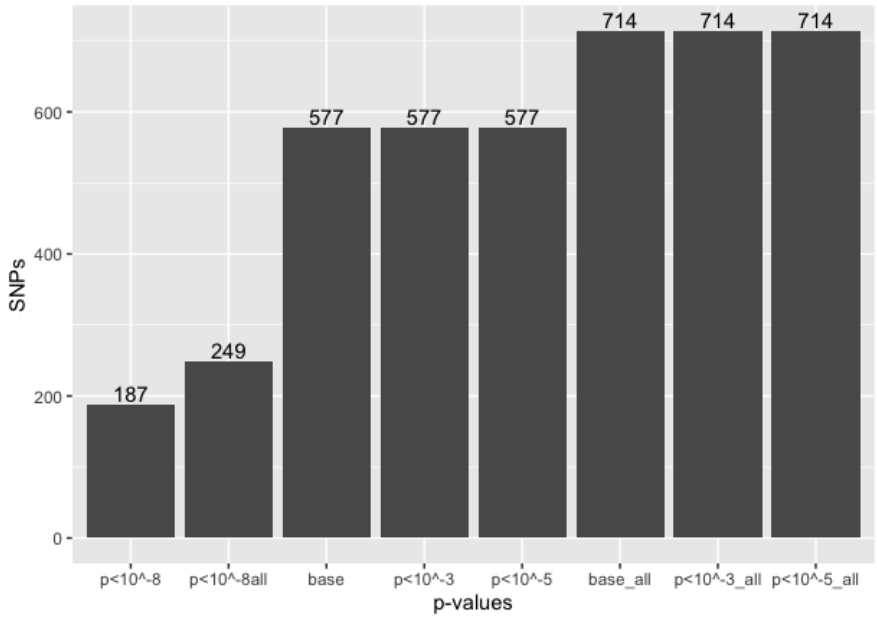
A continuación, el siguiente paso fue analizar el catálogo de GWAS con el término "CLL", para comprobar que nuevos estudios existían desde la realización del estudio previo; es decir, todos los estudios posteriores a la primera mitad del año 2017.

Filtrado por p-valor

Una vez comprobado que habían aparecido numerosos estudios conteniendo el término a partir de dicha fecha, descargamos dichos estudios filtrados por la fecha y comprobamos los p-valores de éstos.

Tras ello, tuvimos que elegir un p-valor de corte para filtrar los resultados obtenidos, para lo que utilizando nuevamente R, hicimos un estudio con los diferentes p-valores que podíamos elegir en función de las variantes obtenidas, filtrando según los valores, según el término buscado en el catálogo, y según si para los SNPs que aparecían en varios estudios, nos quedábamos únicamente con el que tuviese el p-valor más bajo, o los explorábamos todos y filtrábamos a posteriori en base a un p-valor fijo.

Podemos observar en las gráficas que todos los SNPs encontrados tienen un p-valor menor que 10^{-5}, ya que el total encontrado coincide entre los resultados sin filtrar, y los que filtran con un p-valor menor a 10^{-5} y 10^{-3} [Figura 9a]. En el caso de la segunda gráfica, el número de variantes encontradas es mucho menor, pero sigue el mismo recorrido de p-valores, al estar todos los SNPs encontrados por debajo de 10^{-5} [Figura 9b]. Por este motivo, la decisión final fue utilizar el filtro de p-valor < 10^{-8}, para quedarnos así con las variantes de los estudios más relevantes, manteniendo también los SNPs que aparecían en más de un estudio.



a)

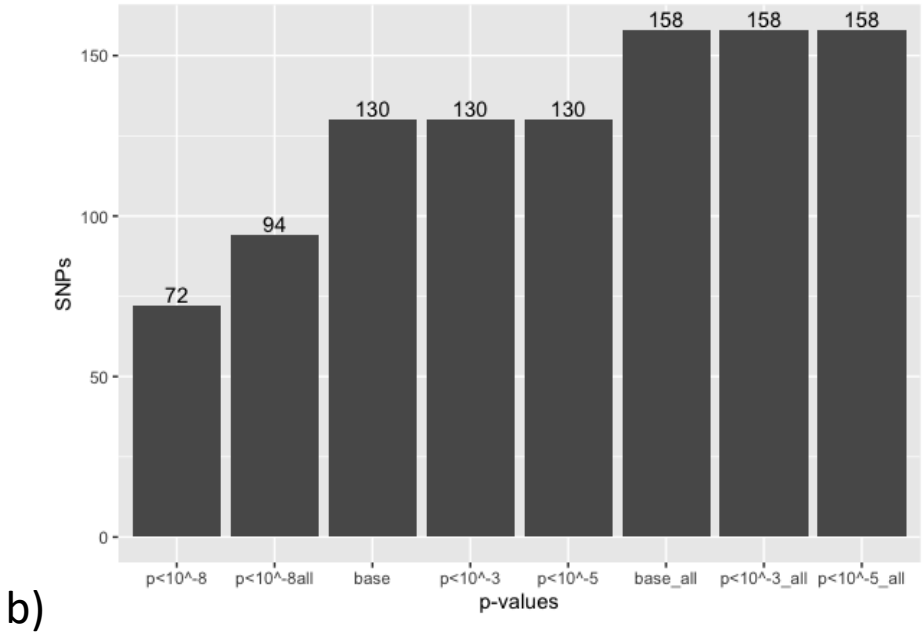


Figura 9: Gráficas que recogen el número de SNPs encontrados en el catálogo de GWAS en base a los diferentes p-valores utilizados para filtrar los resultados, y según si cuando existen SNPs que aparecen en más de una ocasión, nos quedamos con todos o sólo con el que tiene el p-valor más bajo. A) SNPs obtenidos de la búsqueda del término '*lymphoma*', en el que el eje x recoge los p-valores, siendo 'base' el equivalente a no filtrar según el p-valor, y siendo aquellos que llevan añadido el sufijo 'all' los que utilizan todos los SNPs aunque aparezcan en más de un estudio, y los que no lo indican, son aquellos en los que nos quedamos con el estudio con el p-valor más bajo. B) Siguiendo el mismo criterio para el eje X y el Y, pero en este caso, para el término '*Chronic Lymphocytic Leukemia'* .

Selección final de variantes

Para obtener el conjunto final de SNPs, el método que planeábamos utilizar para juntar todas las variantes de las diferentes bases de datos, era mediante la intersección. En el catálogo de GWAS ya habíamos encontrado el p-valor por el que filtrar los resultados, y analizando DisGeNET, encontramos que existía un valor (SCORE VDA, del que ya hablamos previamente) que permitía organizar y filtrar las variantes, de forma similar a como se hacía en el catálogo de GWAS, y además existían otros valores (DSI, DPI) que se basaban el número de enfermedades en las que se manifestaba cada variante, que no era tan interesantes para nuestro experimento. Las variantes en esta base de datos se definen según el identificador de la base de datos dbSNP de NCBI [41] (rs o RefSNP).

En el caso de ClinVar, los identificadores están compuestos por tres letras y nueve números. Las letras indican el tipo (*submitted, reference o variation*), y el número la variante. En cambio, cuando comenzamos a observar los demás campos de ClinVar para decidir cómo proceder con la intersección, encontramos que esta base de datos no permite filtrar, ni existe ningún p-valor o métrica que permita organizar las variantes. Por este motivo, y ya que con las dos otras bases de datos podíamos obtener datos suficientes, decidimos no contar con ella.

Tras tomar esta decisión, utilizamos R para generar un fichero CSV en el que se encontraba la intersección entre las variantes de DisGeNET y GWAS Catalog, que se usaría más tarde.

Bucles

Una vez obtenidas las variantes, el siguiente paso en el proceso para estudiar la predisposición genética heredada de LLC [42] y encontrar los genes implicados, correspondía a la detección de bucles mediante las herramientas denominadas *loop callers*, utilizando para ello datos de tipo Hi-C, que se encuentran disponibles en los servidores del *Barcelona Supercomputing Center*.

La primera tarea de esta fase fue realizar un compendio de las herramientas de análisis de Hi-C existentes en el mercado, tanto las más recientes como las ya establecidas en el mercado, para comprobar cuales se ajustaban más a nuestras necesidades y estudiar sus características [Tabla 1].

Finalmente nos quedamos con los diez candidatos que más acertados parecían para las tareas que pretendíamos analizar. De cada uno de ellos, indicábamos su artículo de respaldo, un pequeño resumen de la herramienta, el enlace a la propia herramienta en sí, un apartado de ventajas y desventajas, y la fecha de publicación, ya fuese de lanzamiento o de la última actualización.

Los candidatos iniciales con los que nos quedamos, para proceder a tomar la decisión de cuales serían los que finalmente utilizaríamos, fueron: HiCeekR [43], HiTC (paquete de BioConductor) [44], GITAR [45], PSYCHIC [46], FastHiC [47], HiCUP [48], PEAKACHU [35], HOMER [49], FitHiChIP [50] y MUSTACHE [34].

De entre ellos, reducimos la lista a la mitad observando las ventajas y desventajas de cada uno, así como analizando lo que permitían realizar y como lo hacían, y los resultados que podíamos obtener tras ejecutarlos. Los elegidos fueron HiCUP, PEAKACHU, HOMER, FitHiChIP.

Finalmente, estudiamos como funcionaba cada uno y cuales eran sus tiempos de ejecución, los ficheros que era necesario utilizar para poder lanzarlos, la reproducibilidad que tenían y la dificultad de instalación, ya que planeábamos ejecutar estas herramientas en los servidores del BSC.

NAME	REFERENCE	SUMMARY	LINK	ADVANTAGES	DISADVANTAGES	DATE OF RELEASE
HiCeekR	Lucio Di Filippo, et al. (2019) *Front. Genet.*	R Graphical User Interface (GUI) that allows researchers to easily perform a complete Hi-C data analysis. With the aid of the Shiny libraries, it integrates several R/Bioconductor packages for Hi-C data analysis and visualization, guiding the user during the entire process	GitHub	-Easy for non-programmers -Hi-C integration with omic datasets -Interactive graphical outputs -Modular structure	-High time-demanding computations -Not provide reproducible research functionalities	04/11/2019
HiTC BioC package	Nicolas Servant, (2021) *Bioconductor*	The HiTC package was developed to explore high-throughput 'C' data such as 5C or Hi-C. Dedicated R classes as well as standard methods for quality controls, normalization, visualization, and further analysis are also provided.	BioConductor	-Extensible framework for Hi-C data -Handles 5C and Hi-C data -Flexible basis for further developments	-If data is not represented in specific ways, it produces high memory consumption and low speed	2020
GITAR	Ricardo Calandrelli, et al (2018) *Genomics, Proteomics & Bioinformatics*	Genome Interaction Tools and Resources (GITAR), a software to perform a comprehensive Hi-C data analysis, including data preprocessing, normalization, and visualization, as well as analysis of topologically-associated domains (TADs)	GenomeGitar	-Opensource -Modular structure -Easy for non-programmers -Reduced storage occupation	-No annotation features -It requires several python libraries to work	1st RELEASE - 12/15 LAST RELEASE - 27/03/20
PSYCHIC	Gil Ron, et al. (2017) *Nature Communications*	Computational approach for analyzing Hi-C data and identifying promoter–enhancer interactions. It uses a unified probabilistic model to segment the genome into domains, which the algorithm then merge hierarchically and fit using a local background model, allowing us to identify over-represented DNA–DNA interactions across the genome.	GitHub	-High identification of over-represented interactions -Identify enriched DNA-DNA interactions -Stronger and sharper enrichment that other methods	'-It needs high resolution Hi-C data -Quadratic running time -3D structure just reflect the set of accesible or active genome regions	27/02/2018
FastHiC	Zheng Xu, et al. (2016) *Bioinformatics*	Approach based on simulated field approximation, which approximates the joint distribution of the hidden peak status by a set of independent random variables, leading to more tractable computation.	UNC	-Fast and accurate to detect long-range chromatin interaction, using simulated field approximation. -Higher peak calling accuracy	-Older and slower than other methods -Weaker interaction enrichment in comparison with newer methods	01/09/2016
HICUP	Steven Wingett, et al. (2015) *F1000Res*	HiCUP is a pipeline for processing sequence data generated by Hi-C and Capture Hi-C (CHi-C) experiments, which are techniques used to investigate three-dimensional genomic organisation. The pipeline maps data to a specified reference genome and removes artefacts that would otherwise hinder subsequent analysis.	Babraham	-Provides statistics summarising each stage of the pipeline -Flexible and publicly available -Compatible with most of analysis tools	-Not really fast -Not in a platform like github -HTML report incomplete	01/09/2016
PEAKACHU	Tarik J. Salameh, et al. (2020) *Nature Communications*	Approach based on simulated field approximation, which approximates the joint distribution of the hidden peak status by a set of independent random variables, leading to more tractable computation.	Github	-Detects high-quality loop interactions from GWAS data -Applicable to train a model with few positive data points -Robust to sequencing depth	-It needs more methods to obtain the complete set of chromatin loops -Depends on chromosome conformation platforms.	07/2020
HOMER	Duttke, S. H., et al. (2019) *Genome Research*	Background model and algorithms, an open-source package, for normalisation and multiple testing that is specifically adapted to CHi-C (Capture Hi-C is a method for profiling chromosomal interactions involving targeted regions of interest, such as gene promoters, globally and at high resolution) experiments.	Homer	-Already used -Hi-C tool that identifies more true-positives than the rest -It can take as input the sole interaction matrix	NOT FOUND	1st RELEASE - 10/12 LAST RELEASE - 03/20
FitHiChIP	Sourya Bhattacharyya, et al. (2019) *Nature Communications*	Computational method for loop calling from HiChIP/PLAC-seq data, which jointly models the non-uniform coverage and genomic distance scaling of contact counts to compute statistical significance estimates.	Github	-Fast and memory efficient -Identifies strogest loops compared to existing methods	-FDR Problems with peaks called not observed in	1st RELEASE - 15/06/15 LAST RELEASE - 08/20
MUSTACHE	Abbas Roayaei Ardakany, et al. (2020) *Genome Biology*	A new method for multi-scale detection of chromatin loops from Hi-C and Micro-C contact maps. Mustache employs scale-space theory, a technical advance in computer vision, to detect blob-shaped objects in a multi-scale representation of chromatin contact maps parametrized by the size of the smoothing kernel.	Github	-Enriched loops and strongly correlated to known signals -Time efficient and does not require GPUs, better reproducibility of loop calls from replicates and provide higher statistical power in comparison with other tools, like HiCCUPSNo annotation features -It requires several python libraries to work	NOT FOUND	1st RELEASE - 15/06/15 LAST RELEASE - 08/20

Tabla 1: Conjunto de las diferentes alternativas de herramientas de análisis de datos Hi-C, incluyendo para cada herramienta el artículo de respaldo, un pequeño resumen, el enlace a la herramienta, ventajas y desventajas y la fecha de lanzamiento y/o última actualización. Marcadas en naranja claro aparecen las opciones más acordes a nuestras necesidades, y en naranja oscuro las herramientas que finalmente seleccionamos para realizar el análisis.

Las herramientas resultantes, que serían con las que trabajaríamos finalmente, fueron MUSTACHE y PEAKACHU, ya que al comprobar todas las características, eran extremadamente rápidas y sencillas de instalar y utilizar, además de tener una mayor reproducibilidad, en comparación con el resto de alternativas, siendo también las herramientas de desarrollo más reciente.

Obtención y preprocesado de los datos

Tras elegir los *loop callers* que íbamos a utilizar para el experimento, la siguiente fase trataba de obtener los datos necesarios para poder ejecutarlos. En este caso, necesitábamos los datos Hi-C (contenidos en ficheros de tipo BAM) para cada una de las líneas celulares que íbamos a utilizar (CLL, MCL, GCBC, MBC, NBC y PBC), que se encontraban previamente procesados en los servidores del BSC, en concreto en el servidor de Nord3.

Todos estos datos genómicos fueron obtenidos gracias al portal de análisis de datos BLUEPRINT (BDAP), (desarrollado a partir de un proyecto de epigenómica internacional; el *International Human Epigenome Consortium),* que a su vez forma parte de EPICO, una plataforma que facilita el acceso a los datos epigenómicos generados por la comunidad científica en multitud de proyectos sin necesidad de disponer de conocimientos técnicos. Por su lado, BLUEPRINT se trata de un proyecto europeo que tiene por objetivo generar y proporcionar a los usuarios epigenomas de referencia [51].

En los servidores del BSC, la ejecución de las principales herramientas bioinformáticas que se utilizaron en el proyecto, como los *loop callers*, se llevó a cabo utilizando la plataforma Singularity [52], que permite realizar virtualización mediante contenedores, aumentando así la reproducibilidad de los entornos en los que se ejecutan estas herramientas, ya que pueden ser completamente copiados y ejecutados en otras plataformas.
Ya obtenidos los datos, y con la ayuda de Singularity, los primeros pasos consistían en preprocesar los ficheros, para lo que, utilizando Samtools, nos encargamos de ordenar los ficheros BAM (comando '*sort'*), para posteriormente hacerles un indexado (comando *'index'*).

Tras realizar estas acciones sobre los ficheros, la siguiente fase era normalizar los datos Hi-C, para que todos estuviesen en la misma escala. Para este fin, utilizamos TADbit, una librería de código abierto para Python [53] que permite, entre otros, analizar y modelar la cromatina en tres dimensiones, así como normalizar estos tipos de datos genómicos.

Para ejecutar este programa, utilizamos la cola de procesos de los servidores del BSC en combinación con entornos ya creados de Singularity específicos para TADbit, y tuvimos que determinar cual sería la resolución que utilizaríamos para normalizar. Observando los parámetros de las herramientas de análisis que utilizaríamos a posteriori, en este caso Peakachu, que proporcionaba unos datos concretos (que veremos a continuación) para utilizar con una resolución de 10kb, establecimos que ésta sería la que utilizaríamos para normalizar los datos genómicos de las seis líneas celulares.

Para poder ejecutar los *loop callers* que habíamos seleccionado, necesitábamos que los ficheros estuviesen en un formato concreto. Para Mustache, teníamos tres alternativas: formato texto, para el que necesitaríamos dos ficheros (archivo de conteo de contactos y archivo de sesgo o *bias*), formato .hic correspondiente a la herramienta de código abierto *Juicer* [54], o formatos .cool y .mcool pertenecientes al paquete *Cooler* [55], que permite almacenar los datos Hi-C de forma escalable.

En cambio, los ficheros de entrada que acepta Peakachu tienen que estar o bien en formato .hic de *Juicer*, o bien en formato .cool de *Cooler*. Por ello, lo más óptimo era elegir uno de los dos formatos para convertir nuestros ficheros .bam ya preprocesados y normalizados, y el más conveniente en nuestro caso, era el formato del paquete *Cooler*, ya que de nuevo, gracias a TADbit, podíamos convertir los ficheros a dicho formato.

Inicialmente decidimos que sería más optimo realizar la transformación por cromosomas, en vez de hacerlo con el genoma completo, ya que de esta manera podríamos procesar los cromosomas independientemente y en paralelo. Para ello utilizamos la función *bin* de

TADbit, y una resolución de 10kb, como habíamos decidido previamente, indicando también que los ficheros se encontraban previamente normalizados. De esta manera, obtuvimos un fichero .mcool por cada uno de los cromosomas (del 1 al 22 y el cromosoma X), y para cada una de las seis líneas celulares.

Antes de realizar la transformación mediante *bin* de TADbit, en una primera instancia intentamos utilizar un conjunto de funciones propias de esta librería, primero *load_hic_data_from_bam()* para cargar los datos Hi-C de un fichero BAM, para posteriormente utilizar la función *write_cooler()* para obtener el fichero cool; sin embargo, el formato de los ficheros generados de esta forma no era válido para utilizarse con Mustache y Peakachu, por lo que terminamos quedándonos con los .mcool que generamos como mencionábamos anteriormente.

Uno de los problemas a los que nos enfrentamos utilizando esta función era a la hora de especificar el cromosoma del que queríamos obtener la información, ya que aceptaba diversos formatos, pero no funcionaba de la misma manera, no siendo capaz de convertir el fichero a *Cooler* en varios casos debido a que si no se realizaba de la manera correcta (p. Ej. *-c 21*) introducía todos los valores en un solo vector, lo que después impedía su correcta conversión. Además, para poder ejecutar esta función, fue necesario instalar la librería h5py dentro del entorno de Singularity, ya que si no TADbit no era capaz de transformar el fichero de salida a .mcool.

Mustache

Ya con los ficheros en el formato que queríamos, divididos por cromosoma y línea celular, el siguiente paso era ya ejecutar los *loop callers*, para lo que habíamos decidido comenzar por Mustache, ya que es una herramienta muy sencilla de instalar, y la preparación de ficheros previa ya estaba realizada.

Mustache (cuyo acrónimo viene de *Multi-scale detection of chromatin loops from Hi-C and Micro-C contact maps)* fue desarrollada el año 2020 como alternativa a otras herramientas clásicas de detección de bucles, basada en métodos de enriquecimiento local para datos de tipo Hi-C y Micro-C de alta resolución. Mustache detecta bucles en la cromatina a resoluciones variadas (aunque en nuestro caso nos centramos en los 10kb) utilizando representación espacial escalar de los mapas de contacto (un reciente avance en la visión artificial). Los píxeles enriquecidos localmente, son los que la herramienta asigna como bucles, utilizando en el proceso un conjunto de filtros seleccionados y diseñados minuciosamente. De esta manera, consigue detectar bucles en la cromatina que, además de ser reproducibles, no son detectables utilizando otras herramientas más antiguas, como HiCCUPS o Fit-Hi-C [34].

Para poder utilizar esta herramienta, los creadores aportan numerosas maneras de realizar su instalación, como PIP, Conda o Docker (indicado en su github). En nuestro caso, la instalamos en los servidores del BSC con ayuda del servicio técnico del grupo utilizando los comandos de PIP (para lo que a su vez es necesario tener instaladas otras dependencias, como una versión de Python igual o superior a la 3.6, numpy, pandas o matplotlib, ya que las utiliza a la hora de realizar el análisis de bucles y devolver los resultados y gráficos).

Como ya disponíamos de los ficheros en el formato necesario y de la herramienta instalada, procedimos a ejecutarla con los parámetros convenientes (indicando los ficheros de entrada y salida, la resolución de 10kb, aunque inicialmente también teníamos la idea de utilizar también una resolución de 5kb, por lo que ejecutamos la herramienta con ambas resoluciones, y una vez por cromosoma y línea celular, con lo que podríamos dejar la herramienta ejecutándose en paralelo dentro del servidor).

Inicialmente, tuvimos problemas con el cromosoma Y, al ser específico masculino y ser muy pequeño, la herramienta generaba errores al salirse las coordenadas genómicas de este cromosoma fuera de los límites establecidos por Mustache, por lo que decidimos quitarlo y trabajar con los 22 cromosomas autosómicos y con el cromosoma X del genoma humano.

Tras esto, obtuvimos los bucles para cada cromosoma dentro de las seis líneas celulares en formato TSV, y conteniendo una línea por cada bucle, indicando el número de cromosoma en el que se encuentra el bucle (el de inicio y el de finalización), las coordenadas de inicio y fin, el FDR (*false discovery rate,* que sería el valor que utilizaríamos posteriormente para filtrar los resultados) y la escala de detección de Mustache para ese bucle.

Una vez obtenidos los bucles de Mustache, tocaba pasar a conseguir los bucles de Peakachu, el segundo *loop caller* que íbamos a utilizar, que a pesar de ser algo más tedioso trabajar con él, era también muy sencillo de instalar y preparar los ficheros para su ejecución.

Peakachu

Su nombre viene de *Unveil Hi-C Anchors and Peaks,* y al igual que Mustache, fue desarrollada en el año 2020. En este caso, esta herramienta utiliza aprendizaje automático para encontrar los bucles de la cromatina en los mapas de interacción a nivel de genoma completo, y esto lo hace definiendo conjuntos de entrenamiento positivos y negativos, con los que después construye modelos para clasificar los bucles [35]. El conjunto positivo pueden ser tanto cualquier lista de interacciones de experimentos biológicamente enriquecidos o de experimentos de imágenes de alto rendimiento, mientras que el conjunto negativo se genera a partir de una muestra aleatoria de dos poblaciones (contactos con distancias genómicas tanto similares como superiores al conjunto positivo) [Figura 7]. Tras esto, utilizando ciertos parámetros definidos por la propia herramienta, busca el mejor modelo de *Random Forest* que separe ambas clases, detectando así los bucles a partir de los mapas de contacto. La mayor ventaja que tiene Peakachu respecto a las herramientas alternativas es su robustez en la profundidad de secuenciación, lo que lo hace válido para predecir bucles en la cromatina en datos Hi-C con pocas lecturas.

En este caso, los creadores de la herramienta nos dan un flujo de ejecución más concreto para proceder con la instalación de Peakachu, teniendo para ello que crear un entorno en Conda con las herramientas necesarias (Python 3.6, scikit, numpy, pandas, h5py o cooler) y posteriormente clonarla e instalarla desde GitHub. De esta forma fue de la que se realizó la instalación de la herramienta en el servidor, siendo necesario además activar el entorno en él para poder utilizar Peakachu.

En este caso, la ejecución del *loop caller* no fue tan directa, ya que investigando en el Github de la herramienta (accesible en https://github.com/tariks/peakachu), descubrimos que no era posible realizar el entrenamiento del modelo *Random Forest* con los datos Hi-C, ya que no disponíamos de un fichero .bedpe con el conjunto del entrenamiento positivo, el cual era necesario para poder realizar esta acción.

Por ello, decidimos probar con otras opciones. La primera fue intentar encontrar el umbral para los bucles diferenciales, utilizando valores de probabilidad y modelos preentrenados, pero terminamos descartándolo por la elevada carga computacional además de no adaptarse a lo que nosotros estábamos buscando.

Debido a esto, finalmente, decidimos optar por la alternativa de utilizar Peakachu como un *loop caller* clásico, para lo que el primer paso era extraer el número total de pares intracromosómicos, con la función *depth* de Peakachu.

Sin embargo, tuvimos muchos problemas con esta función, ya que siempre fallaba al intentar encontrar el objeto 'chroms' dentro de nuestros .mcool, por lo que comenzamos a trabajar directamente con el paquete h5py, para investigar como podíamos abrir los ficheros, y obtener las claves necesarias.

Para ello, en otro intento de resolverlo, contrastamos nuestro .mcool contra el fichero de ejemplo de Peakachu, y comprobamos que tampoco funcionaba correctamente, ya que este no tenía etiquetas.

Finalmente, concluimos que el problema venía directamente de su GitHub, ya que en la guía de uso de la herramienta, la función *depth*, indicaba un comando con una pequeña falta, que no permitía que se ejecutase correctamente. Simplemente corrigiendo este detalle, conseguimos que funcionase sin problema y detectase nuestros ficheros .mcool. A continuación, se informó a los desarrolladores para que corrigiesen el error de su página.

Lanzamos la función *depth* dentro de un bucle en bash para todos los cromosomas que teníamos, incluido el X, obteniendo las lecturas indicadas en la figura [Figura 10]. Con esta información, calculamos la media de todas las lecturas (421859075.5), que posteriormente dividimos a la mitad al ser todas las lecturas dobles (debido a la indexación por coordenadas, que duplica el número de líneas de los ficheros bam), para así obtener el punto de corte que necesitábamos para elegir el modelo para predecir los bucles de los que proporcionan los creadores de Peakachu en su Github, en este caso utilizando H3K27ac HiChIP (ya que se trata de un modelo más funcional, se asocia, entre otros, a promotores) como conjunto de entrenamiento. Obtuvimos un total de aproximadamente 200 millones de lecturas, por lo que escogimos el modelo H3K27ac a 10% para realizar la obtención de los bucles.

En este punto fue donde decidimos utilizar definitivamente una resolución de 10kb para todos los procesos, ya que los modelos proporcionados por Peakachu para realizar las predicciones, únicamente se encontraban a dicha resolución, por lo que finalmente descartamos los ficheros que habíamos generado con resolución de 5kb y trabajamos con los de 10kb para el resto del proceso.

A continuación, nos planteamos realizar todo el proceso de nuevo, pero esta vez utilizando el genoma completo en vez de dividirlo por cromosoma, para comprobar si existían diferencias y cuales eran éstas; sin embargo, aunque la conversión inicial a *Cooler* no fue extremadamente larga, a partir de ahí el resto de procesos tardaban varios días, como la ejecución de los *loop* callers, y ni si quiera en los servidores del BSC era posible ejecutar la mayoría, ya que superaban los tiempos de ejecución máximos permitidos por sus directrices.

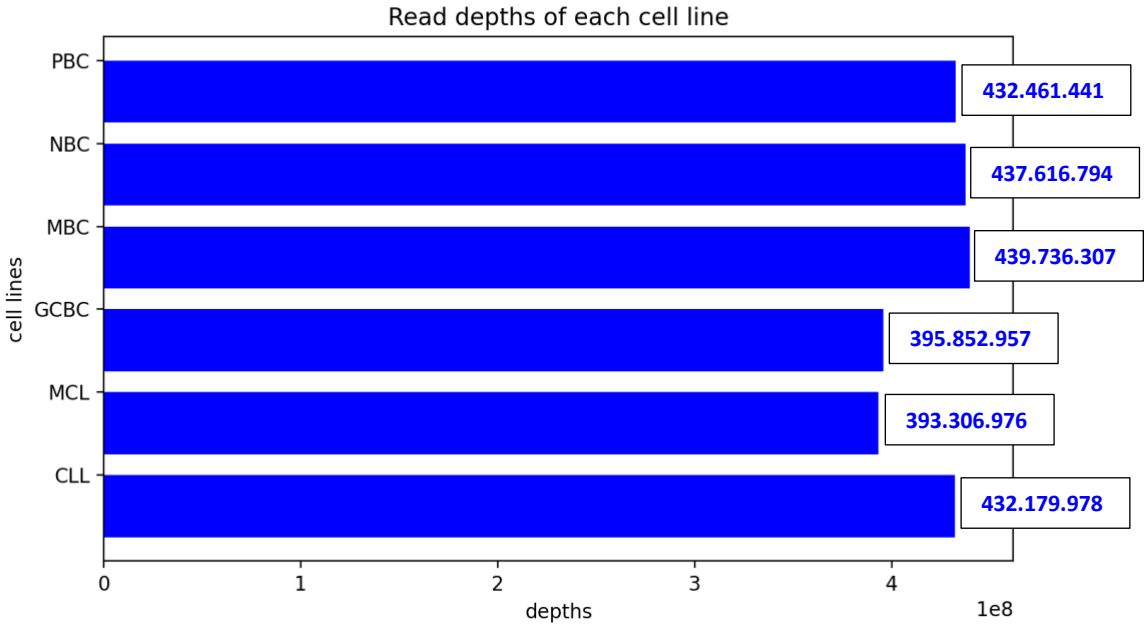


Figura 10: Gráfica obtenida mediante Python del número total de lecturas para cada una de las líneas celulares, calculadas utilizando la función *depth* de Peakachu, que posteriormente utilizamos para calcular el porcentaje del modelo que debíamos utilizar para la predicción

La siguiente fase implicaba ya la obtención de los bucles predichos por la herramienta, para lo cual, el primer paso implicaba calcular la probabilidad de interacción por píxel para cada cromosoma utilizando la función de Peakachu *score_chromosome*, indicando en su ejecución la resolución y el fichero del modelo H3K27ac que habíamos descargado previamente. Esta función, nos generó un fichero tipo BED para cada cromosoma, y para cada una de las líneas celulares.

El siguiente paso, era obtener los bucles detectados por Peakachu con la función *pool* aplicada sobre cada fichero .bed de cada cromosoma. Para ello, nuevamente ejecutamos un bucle en bash para obtener un fichero por cromosoma, según el umbral (o *threshold*) utilizado, y siguiendo lo recomendado por el Github de la herramienta, creamos este bucle utilizando un umbral de 0.9; sin embargo, calculamos el número de bucles que había obtenido Peakachu para cada línea celular y cromosoma, y muchos de ellos se encontraban vacíos, por lo que decidimos probar a reducir el valor del umbral, para que éste fuese más laxo y poder obtener así más bucles.

Cambiando el valor del umbral, íbamos obteniendo diferentes números de bucles en total para cada línea celular, por lo que finalmente decidimos crear un conjunto de bucles

encargados de ejecutar en paralelo para cada cromosoma y línea celular la función *pool* de Peakachu con diversos umbrales (los elegidos fueron 0.1, 0.2, 0.3, 0.4, 0.5 y finalmente, uno intermedio y más alto, 0.7, ya que a partir de ese prácticamente no se encontraban bucles en la mayor parte de los casos).

De esta manera, obtuvimos todos los bucles en varios ficheros, que son con los que trabajaríamos a continuación para decidir cual iba a ser el *threshold* final que utilizaríamos como corte para escoger los bucles detectados con los que nos quedaríamos para realizar la posterior intersección y análisis final de resultados.

Jaccard Index de los loop callers

Así, nuestra siguiente tarea era escoger cual iba a ser el filtro o punto de corte para elegir los bucles, para lo que necesitábamos analizar, en el caso de los bucles detectados con Mustache, que FDR utilizar, y en el caso de Peakachu, que umbral sería el más óptimo.
Para ello, decidimos utilizar el índice Jaccard, que es una medida de la similitud entre dos conjuntos, y que nos mostraría de manera clara y directa que filtro sería más óptimo utilizar en cada caso, para lo que era necesario calcular, siguiendo la definición del índice, la intersección entre dos conjuntos dividida por su unión. Los conjuntos que íbamos a enfrentar en este caso, iban a ser las diferentes líneas celulares, dentro de cada uno de los valores diferentes de umbral, en el caso de Peakachu, o de FDR, en el caso de Mustache, buscando así maximizar tanto el número de bucles encontrados como el valor del índice de Jaccard (que tomaría valores de entre 0 y 1).

Sin embargo, con el fin de poder utilizar los ficheros directamente para obtener estos valores, el primer paso fue juntar todos los bucles de todos los cromosomas, ya que se encontraban separados por cromosoma y línea celular (esta última división la mantendríamos para poder realizar las comparaciones). Para realizar esto, utilizamos un script simple realizado en Bash.

Más tarde, el siguiente paso fue separar los ficheros según el filtro. En el caso de Peakachu, ya los teníamos divididos según el umbral (entre 0.1 y 0.5 y, además, 0.7), pero en el caso de Mustache estaban todos juntos, por lo que creamos un script, en este caso combinando Python con Bash, para generar cinco ficheros por cada línea celular, ya que habíamos decidido comprobar el índice de Jaccard en cinco intervalos de FDR diferentes, que serían entre 0.1 y 0.5, entre 0.05 y 0.1, entre 0.01 y 0.05, aquellos con FDR menor que 0.01, y aquellos que fuese menor que 0.05.

Comenzamos tratando de calcular este índice con los ficheros de Peakachu, al tenerlos ya previamente separados según el umbral. Inicialmente, creamos un código que era computacionalmente demasiado pesado (numerosos bucles anidados), ya que no nos permitía obtener fácilmente los índices, especialmente utilizando los umbrales más pequeños (0.1 y 0.2), ya que el número de bucles detectados era muy alto. Por este motivo, decidimos modificar el código y crear uno que fuese válido para analizar los ficheros de ambas herramientas. Para ello, el primer paso, era formatear los ficheros.

El objetivo de esta acción era únicamente quedarnos con las coordenadas genómicas de cada bucle, quitando el resto de métricas de cada fila, de forma que calcular el índice fuese algo directo y fiable, tanto en el caso de Mustache como en el de Peakachu.

Tras ello, calculamos el índice de Jaccard para ambas herramientas, enfrentando en gráficas el número de bucles y el índice, para cada una de las intersecciones entre las líneas celulares, enfrentándolas una con otra hasta completar todas las combinaciones, y en el caso de Mustache, organizándolas según cada uno de los intervalos de FDR, y en el caso de Peakachu, según cada umbral utilizado.

Para poder dibujar lo anteriormente mencionado, fue necesario utilizar una gráfica de doble eje vertical de Python, perteneciente a la librería *matplotlib*. El principal problema derivó de que los umbrales mas pequeños y FDRs más grandes, correspondían a un grandísimo número de bucles, por lo que tuvimos que procesar toda esa información en los servidores del BSC utilizando numerosas CPUs, para posteriormente poder dibujarla.

Como podemos observar en la gráfica de los FDRs de Mustache, a medida que disminuye el umbral del FDR, el número de bucles encontrados por la herramienta también disminuye, mientras que los índices se mantienen bastante constantes. Por este motivo, hemos elegido utilizar los bucles encontrados con un FDR por debajo de 0.05, al ser un valor de FDR bastante bajo (la mitad que en el caso anterior) y ya que es donde encontramos un índice mayor para la mayoría de líneas celulares, manteniendo un número total de bucles asequible (entre 4000 y 6000) [Figura 11].

En el caso de la gráfica de los umbrales de Peakachu, el punto donde se maximizan tanto el índice como el número de bucles es cuando el umbral es igual a 0.2, ya que luego el número de bucles detectado (que cómo podemos comprobar, con este umbral es muy superior al total de bucles detectado por mustache) disminuye en gran medida. De hecho, con los valores más altos (0.4 - 0.7), no se encuentra ningún bucle. Aunque el valor del índice de Jaccard aún no alcanzando su valor máximo con este valor del umbral, es bastante asequible, similar al de mustache, por lo que los bucles detectados por Peakachu dentro de este umbral fueron los elegidos para realizar la intersección con las variantes y seguir con los análisis [Figura 12].

Ya obtenidos los bucles elegidos de cada una de las herramientas, el siguiente paso conllevaría realizar la intersección de los bucles con las variantes, como habíamos indicado previamente en el flujo del proyecto.

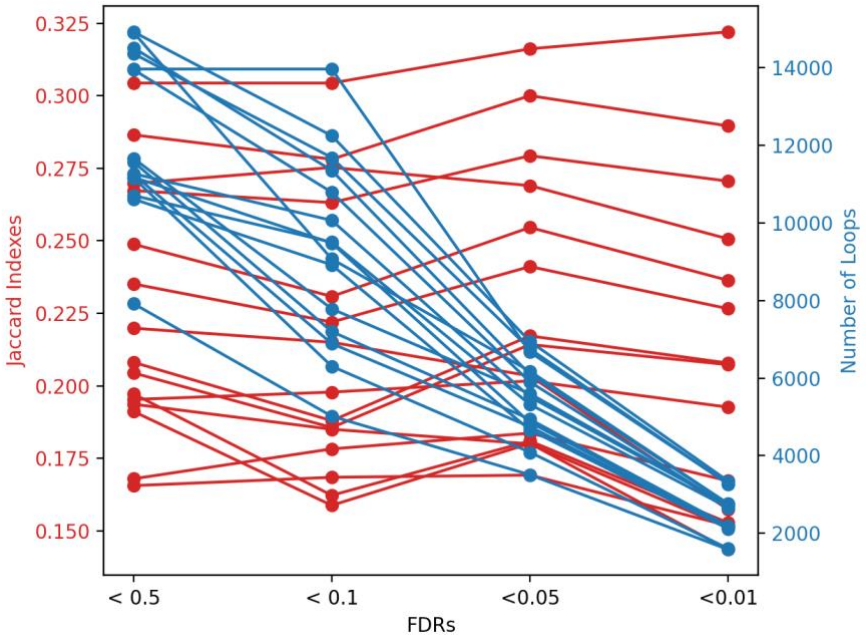


Figura 11: Índices de Jaccard y número total de bucles para cada intersección de cada línea celular con el resto, organizado según el FDR, proveniente del *loop caller* Mustache. Los intervalos de FDR utilizados se indican en el eje X de la gráfica. Cada línea representa la comparación de un par de líneas celulares.

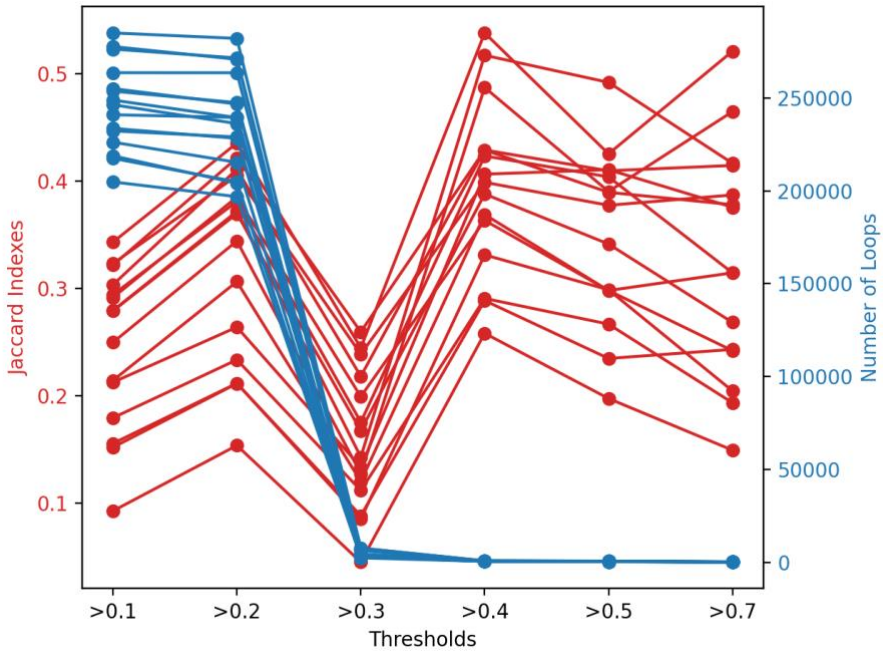


Figura 12: Índices de Jaccard y número total de bucles para cada intersección de cada línea celular con el resto, organizado según el umbral o *threshold* utilizado, proveniente del *loop caller* Peakachu. Los diferentes umbrales se indican en el eje X, del 0.1 al 0.5, y además el 0.7. Cada línea representa la comparación de un par de líneas celulares.

Pero antes, teníamos que decidir si utilizaríamos los bucles encontrados únicamente en ambas herramientas (intersección) o, por el contrario, el total de bucles de ambas herramientas, buscando qué variantes caen en los bucles de cada herramienta por separado, para posteriormente encontrar las interacciones entre bucles y variantes. Esta decisión dependería del total de bucles encontrados y del índice de Jaccard de la intersección.

Para ello, nos encargamos de encontrar esta intersección y calcular el índice de Jaccard, para lo que en este caso, enfrentamos los bucles encontrados en cada línea celular por cada herramienta, con los umbrales definidos en el paso anterior [Figura 13].

Como podemos observar, los valores de ambos ejes son muy bajos, siendo el mayor Jaccard encontrado de 0.024 para la línea celular NBC, y el mayor número de bucles de aproximadamente 2300, en las líneas MCL y MBC, por lo que no nos serviría para seguir adelante con el estudio, ya que el número de resultados sería demasiado bajo, con lo que sería complicado sacar conclusiones al respecto.

Por ello, decidimos utilizar el conjunto de bucles provenientes de ambas herramientas, en las que la mayor parte de bucles provenían de Peakachu (sobre 250000), mientras que de Mustache sólo obtuvimos unos 6000 bucles, como podemos comprobar en las figuras 12 y 13.

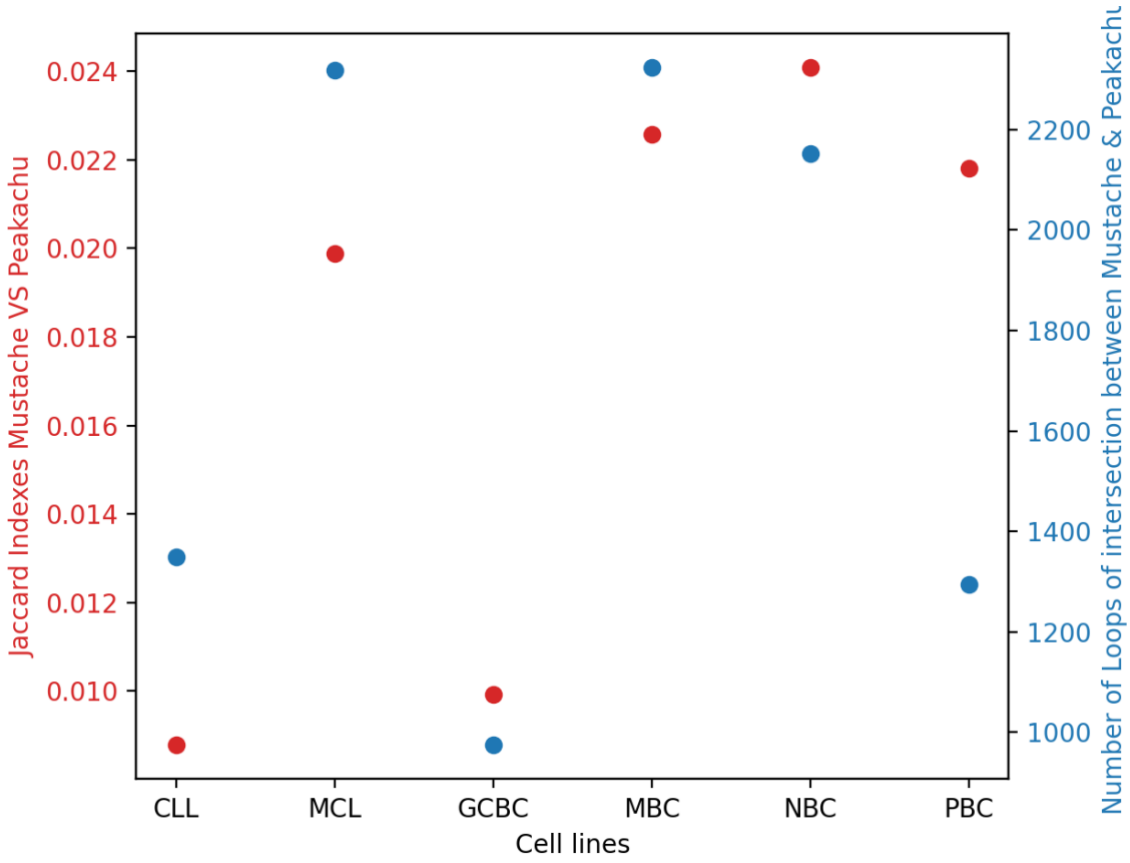


Figura 13: Índices de Jaccard y número total de bucles para la intersección entre los dos *loop callers* utilizados (mustache y peakachu) por cada línea celular.

Intersección entre bucles y variantes

Por lo tanto, el siguiente paso sería buscar la intersección del conjunto de variantes elegidas al comienzo del proyecto con ambas herramientas de análisis, Mustache y Peakachu, por separado, para posteriormente unir todas las interacciones encontradas en un solo fichero

tipo CSV (*LoopVariants.csv*), que sería el fichero que utilizaríamos para encontrar los resultados finales y realizar el análisis para así obtener las conclusiones necesarias.

Los ficheros que utilizaríamos para ello serían los que mencionamos previamente, para Peakachu aquellos con un umbral mayor a 0.2, y para Mustache, los que su FDR se encontraba por debajo de 0.05. Transformamos los nombres de los ficheros para indicar la línea celular y la herramienta de la que provenían, y los desplazamos a la carpeta destino que posteriormente sería el origen de los datos.

Para realizar la intersección, utilizamos la herramienta RStudio, junto con un grupo de librerías (instaladas con BioConductor), con la que generamos una función encargada de intersectar los bucles con las variantes, especificando además la línea celular y la herramienta (que se obtendría directamente del nombre del fichero), así como las kilobases (kb) que se extenderían en cada extremo de las regiones de contactos de la cromatina (bucles). Tras ello, utilizamos esta función con todos los bucles de Mustache por un lado y Peakachu por otro, por cada línea celular, y fuimos escribiendo en una nueva tabla cada bucle que contenía una variante. Finalmente, juntamos todos los bucles en un *dataframe* global y escribimos toda la información en un único fichero, *LoopVariants.csv*.

A continuación quisimos comprobar el número de intersecciones encontradas entre bucles y variantes en el paso anterior, para lo que creamos un histograma, que nos ayudaría a calcular este dato por cada línea celular [Figura 14]. Como se puede observar, el mayor número de interacciones se encuentran en la línea celular LLC, con casi 12000, seguida de cerca por MCL y después por MBC, con valores también bastante altos. La línea con menor número de interacciones es PBC, con casi 6000, pero manteniéndose aún así en un número total de interacciones bastante elevado, por lo que consideramos que el fichero que contenía todas estas interacciones era coherente y aceptable para poder seguir adelante con el estudio, al existir un número alto de interacciones en todas las líneas (entre 6000 y 12000, como mencionábamos), lo que pensamos en una primera instancia que nos permitiría sacar conclusiones al respecto en los pasos siguientes.

Anchura de los puntos de anclaje de bucles y solapamiento con variantes

Habiendo ya obtenido el fichero con las intersecciones entre bucles y variantes, y habiendo comprobado que los datos obtenidos eran asequibles, ya podíamos pasar a los pasos finales, pero decidimos hacer unos análisis previos a esto, como indicábamos en el esquema del flujo del experimento, para lo que íbamos a representar algunas relaciones utilizando gráficas de R.

Atendiendo a la incertidumbre en kilobases extendidas en el punto de anclaje del bucle que realizamos (0-50), nuestro objetivo fue representar gráficamente las diferentes relaciones entre los SNPs únicos en los bucles, y los propios bucles, según las extensiones.

Para ello, realizamos tres gráficas representando tres relaciones diferentes:

- Número total de bucles versus extensión en kb de las regiones de los bucles [Figura 15a].
- Número de SNPs únicos versus extensión en kb de las regiones de los bucles [Figura 15b].
- Proporción de SNPs únicos provenientes de la base de datos del GWAS Catalog respecto del total VS KB extendidas en las regiones de los bucles [Figura 15c].

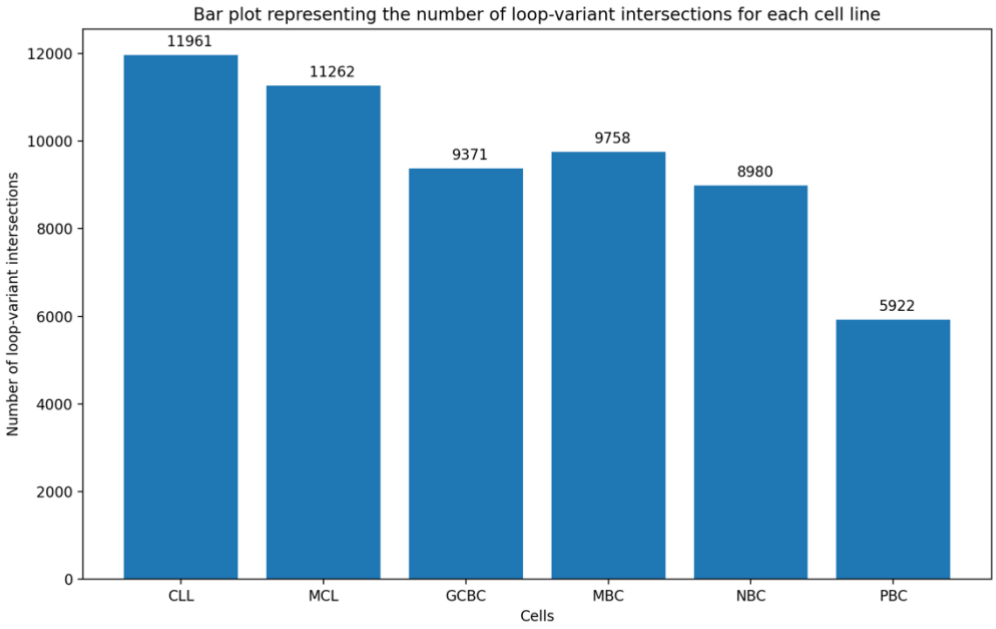


Figura 14: Diagrama de barras en el que se indica el número de interacciones bucle-variante existentes para cada línea celular en el fichero obtenido en pasos anteriores. En el eje horizontal se indica cada línea celular, y en el vertical el número total de bucles.

En la primera de ellas, podemos observar que, como era de esperar, la frecuencia de aparición de bucles aumenta de forma directamente proporcional al aumento de las regiones de anclaje. De esta forma, observamos un crecimiento continuo de unos 3000 bucles por cada 10kb extendidas, empezando sobre unos 2000 bucles al no realizar esta extensión, hasta llegar a aproximadamente 17000 bucles cuando la extensión es de 50 kilobases.

En la segunda gráfica, se observa que el número de SNPs encontrados aumenta en gran medida al extender las primeras 10 KB, ya que la mayor parte de los SNPs aparecen sin realizar dicha extensión, y muy pocos son encontrados con el resto de aumentos, a diferencia de los bucles.

Sin embargo, al comprobar la última gráfica, que analiza la proporción de polimorfismos provenientes del catálogo de GWAS respecto al total de SNPs encontrados, deja entrever que la mayor parte de SNPs que aparecen sin realizar ninguna extensión pertenecen al GWAS Catalog, disminuyendo esta proporción a medida que aumenta la cantidad de kilobases extendidas, debido a que los SNPs que aparecen, especialmente en las dos primeras fases (10kb y 20kb) pertenecen a la base de datos DisGeNET. Esta proporción aumenta ligeramente de nuevo en las siguientes fases (30kb, 40kb y 50kb), a pesar de no

alcanzar los valores existentes sin extender ninguna base. A pesar de ello, el número de SNPs provenientes del GWAS Catalog, es siempre mayor a las variantes encontradas en DisGeNET.

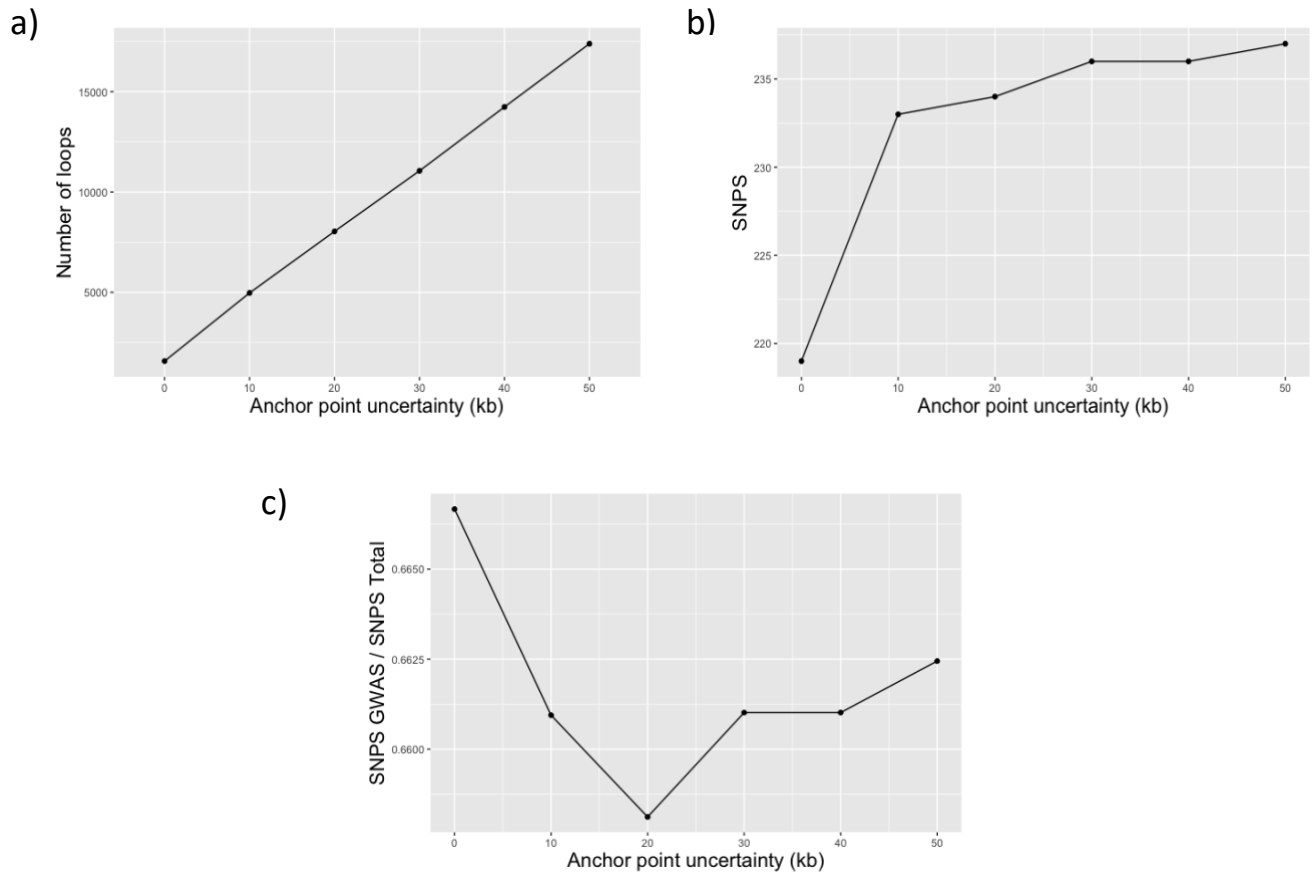


Figura 15: Gráficas representando las relaciones existentes entre los SNPs únicos y los bucles encontrados, atendiendo a la incertidumbre en KB extendidas en las regiones del punto de anclaje de los bucles, realizadas con el lenguaje de programación R. a) Relación entre el número de bucles y las KB extendidas en el punto de anclaje, b) Relación entre el número de SNPs únicos y las KB extendidas en el punto de anclaje, c) Relación entre la proporción de SNPs únicos provenientes del catálogo GWAS respecto del total y las KB extendidas en el punto de anclaje.

Tras analizar las gráficas en conjunto, decidimos que lo más conveniente para seguir con el procedimiento era utilizar una incertidumbre de 0 KB extendidas en el punto de anclaje de los bucles, ya que, como se puede observar, al aumentar esta incertidumbre, el número de SNPs detectados crece muy poco (de 220 en 0 a 237 aproximadamente en 50), mientras que, el número de bucles aumenta mucho más (de 3000 a 17000, como mencionábamos antes) lo que nos generaría un exceso de ruido en las siguientes fases del experimento. Además, la proporción de SNPs procedentes de GWAS alcanza su máximo en 0KB, que sin implicar que sean mejores, si que aportan una mayor cantidad de información.

Anotación de regiones complementarias y contexto genómico

Por último, previo a la generación de las gráficas finales, era necesario anotar las regiones complementarias. Para ello, utilizamos nuevamente R y algunos paquetes de *Bioconductor* que nos ayudarían en el proceso, como *GenomicRanges, annotatr,* o el paquete *TxDb.Hsapiens.UCSC.hg38.knownGene*, que pone a nuestra disposición bases de datos de

anotaciones del genoma humano hg38 del Consorcio de Referencia del Genoma (GRC), en combinación con el paquete *org.Hs.eg.db*, que nos proporciona anotaciones del genoma humano (tipo *genome-wide*).

De esta manera, generamos un programa encargado de leer la tabla de intersecciones de bucles y variantes (*LoopVariants.csv*), para después anotar tanto la región del bucle con la variante como la región del bucle complementaria a la variante. Una vez hecho esto, anotamos las regiones sin extender los extremos (0KB), como habíamos decidido previamente, gracias a las gráficas, utilizando el paquete *annotatr* mencionado previamente, para finalmente escribir una tabla con las regiones del bucle donde caen las variantes, y otra con las regiones complementarias, que son utilizadas para generar una tabla con las anotaciones complementarias y utilizarla para las gráficas finales, analizando en este proceso cual es el contexto genómico de las regiones complementarias y en cuantos bucles aparece cada una de las variantes que hemos obtenido previamente, dentro de LoopVariants.csv, con 0KB extendidas.

Una vez obtenidas las anotaciones del contexto genómico de las regiones complementarias, el siguiente paso fue generar un conjunto de gráficos indicando diferente información para cada una de las variantes: el número de veces que aparecía cada variante en un bucle, el p-valor, *odds ratio* o *RAF* (*risk allele frequency*), estos últimos presentes únicamente en las variantes provenientes de GWAS, y finalmente el contexto genómico tanto de las propias variantes, como de las regiones complementarias en los bucles de cada una de las líneas celulares. Estos gráficos fueron realizados basándose un script presente en el trabajo realizado por Paula Ballcels, adaptándolo a nuestros datos (disponible en: https://github.com/bsc-life/epintegrate) [Figura 17].

Con estos últimos datos, sería con los que buscaríamos en que momentos surgiría el cambio del contexto genómico de una variante respecto a un bucle. Principalmente, consultamos las variantes que eran intrones, intergénicas o regiones reguladoras (como 3'-UTR), y por el lado de los bucles, buscamos aquellas que principalmente fuesen de tipo promotor, y además las de tipo *1to5KB*, las cuales indicaban que no se encontraba exactamente en el promotor, pero sí en las siguientes kilobases, concretamente en el intervalo de 1 a 5 KB desde el promotor, ya que los promotores serían los más relevantes para analizar la incidencia de las variantes en LLC.

Resultados

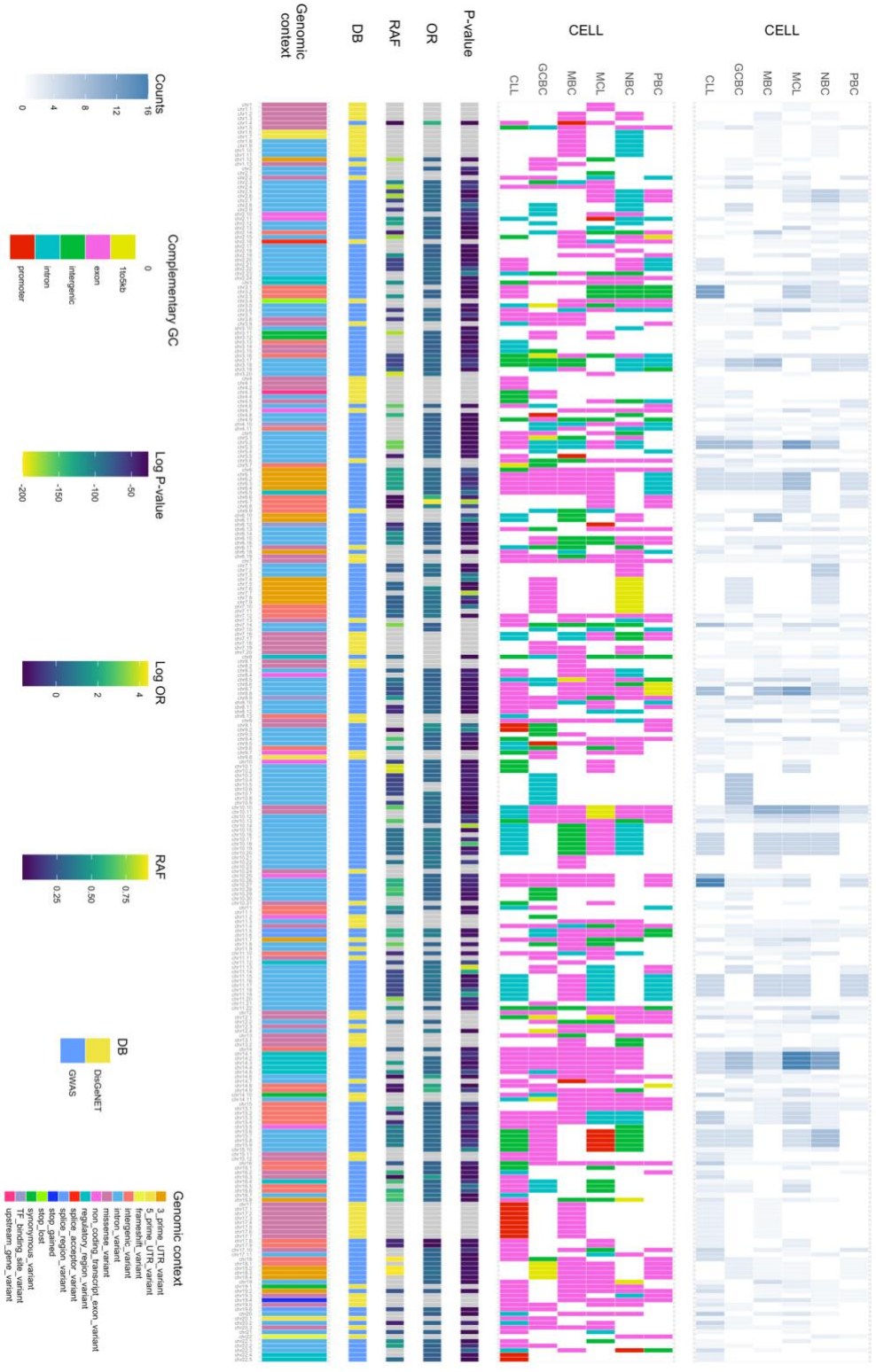


Figura 16: Gráfica combinada en la que se muestran todas las variantes encontradas, el contexto genómico en el que se encuentran, la base de datos de la que provienen, su p-valor, *odds-ratio y risk allele frequency* (estas tres últimas sólo para las variantes provenientes de GWAS Catalog), el contexto genómico de la zona complementaria en la que influyen en cada una de las líneas celulares analizadas, y finalmente el número de bucles en los que se encuentra cada variante.

Gracias al gráfico combinado que obtuvimos [Figura 16], teníamos la capacidad de analizar qué variantes serían las mas influyentes. Para ello, observamos las variantes de LLC que se encontrasen situadas, principalmente, en introns (pero también algunas cuyo contexto genómico era intergénico o 3'-UTR), pero sobretodo comprobamos que en la zona complementaria, entrase en contacto o bien con un promotor, o bien con una zona cercana a éste (*1to5KB*). Esta categoría indicaba que no se encontraba exactamente en el promotor, pero sí en las siguientes kilobases, concretamente en el intervalo de 1 a 5 KB desde el promotor, ya que los promotores serían los más relevantes para analizar la incidencia de las variantes en LLC, y comprobamos que el p-valor fuese aceptable, así como los valores de *Odd Ratio* (OR) y *Risk Allele Frequency* (RAF) (en el caso del p-valor y el OR, utilizamos el logaritmo negativo por diversas razones, como la gestión de números muy pequeños por parte de los equipos informáticos, o situar los valores de forma gráfica en la tabla con números de un tamaño asequible para ser representados). Esto sólo era posible para las variantes provenientes de GWAS, ya que es la única base de datos de las utilizadas que nos proporciona dicha información, pero afortunadamente, la mayor parte de las variantes seleccionadas y analizadas estaban presentes en el GWAS Catalog.

En este caso, las variantes aparecen anotadas con nombre del cromosoma seguido de su índice, y posteriormente un punto y el número de variante presente en dicho cromosoma (por ejemplo, el cuarto SNP encontrado en el cromosoma 7 sería de la forma 'chr7.5', ya que el primero de cada cromosoma se representaría como el 0, pero sin indicarse, siendo 'chr7').

Posteriormente, utilizando un script en R y el buscador de genes de la Universidad Santa Cruz, de California (UCSC Genome Browser), comprobamos cada una de las coordenadas genómicas de las variantes que habíamos seleccionado de las casi 300 de las que disponíamos inicialmente, siguiendo el criterio del contexto genómico explicado previamente (intron-promotor, etc), y asociamos cada variante a uno o varios genes dentro del genoma humano (GRCh38/hg38). Una vez obtenidos estos genes, 27 concretamente, anotamos cada uno de ellos con su correspondiente variante, para realizar un *Gene Set Enrichment Analysis* [Tabla 2].

Para este paso, una vez teníamos la lista de genes, utilizamos la herramienta Enrichr (accesible aquí), para el análisis de enriquecimiento genético. Obtuvimos diversos tipos de enriquecimiento, pero nos centramos en aquellos que tuviesen relación con el cáncer, o más concretamente, con la leucemia, o de ser posible, con LLC.

Para ello, Enrichr [72] divide los enriquecimientos por tipo (transcripción, *pathways*, ontologías, enfermedades…), y a su vez según de que origen obtiene la información. En nuestro caso, priorizamos aquellos enriquecimientos organizados por enfermedades y *pathways*.

En el caso de los pathways de *Elsevier* [73], se encontraban enriquecidas la leucemia linfoblástica aguda de células B y un subtipo de linfoma, por lo que comprobamos que genes se encontraban enriquecidos [Figura 17a]. De la misma manera, en el apartado de enfermedades, observando la información proveniente del GWAS Catalog 2019 y de enfermedades raras de GeneRIF [74], también vimos que se encontraban enriquecidas

varios tipos de leucemia, incluyendo LLC, por lo que comprobamos también que genes se encontraban enriquecidos para cada tipo, utilizando la herramienta *Clustergram* [75] de Enrichr [Figura17b y Figura 17c].

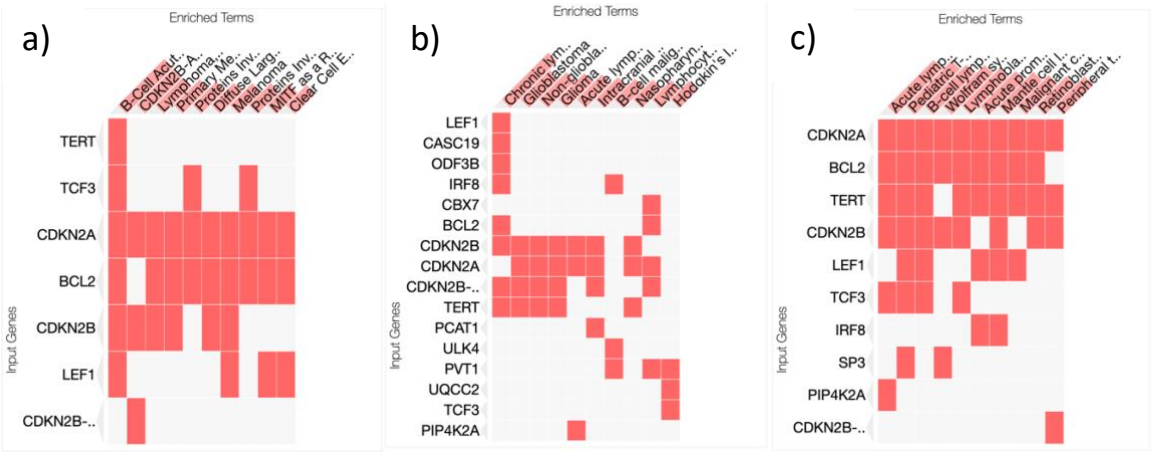


Figura 17: genes enriquecidos relacionados con diferentes tipos de leucemias y linfomas, incluyendo LLC, obtenidos mediante la herramienta Enrichr al introducir los genes que hemos obtenido mediante las coordenadas de las variantes seleccionadas. A) Genes y enfermedades enriquecidas en la colección de patways de *Elsevier*. B) Genes y enfermedades enriquecidas en el catálogo GWAS 2019. C) Genes y enfermedades enriquecidas en las listas de genes y enfermedades raras de GeneRIF.

Posteriormente, el siguiente paso era concatenar los genes que habíamos encontrado enriquecidos en cada uno de los casos, priorizando en el proceso aquellos que se repitiesen en más de un caso (como los genes *TERT* o *TCF3*) y los que estaban presentes directamente en LLC (como los genes *LEF1* o los genes de la familia *CDKN2-*). Tras esto, hacer un análisis de cada uno de ellos, y ver cuáles influían en cáncer, cuáles lo hacían en mayor medida, y de que forma lo hacían (inhibición/supresión, activación, etc.), en base a los estudios realizados en otros trabajos sobre las funciones e influencias de cada uno de estos genes.

Sin embargo, previo a realizar este análisis, observamos un comportamiento peculiar en los genes presentes en el análisis de enriquecimiento genético, ya que en los genes que aparecían en el catálogo GWAS enriquecidos en LLC, también existían mutaciones dentro de ellos; por lo tanto, comprobamos cuáles de los genes que habíamos obtenido del *Genome Browser* mediante las coordenadas, tenían algún SNP en LLC. Para ello, anotamos las posiciones de inicio y fin de cada uno de estos genes dentro de los cromosomas del genoma humano, y a continuación utilizando un script de *Python*, comprobamos que SNPs de los que habíamos escogido previamente, se encontraban dentro de alguno de estos genes, y ver cuáles se quedaban fuera de los genes a los que podían estar afectando. Para comprobar esto, anotamos toda esta información en una tabla, incluyendo las posiciones de inicio y final de los genes a los que parecían estar afectando las variantes, así como la cadena (+ o -) [Tabla 2].

Nº Chr	Coordenada	CG variante	CG comp	Linea celular	Gen	Coord Inicio	Coord Final	Cadena
Chr 2	173.944.171	intron	1to5KB	PBC	SP3	173,900,773	173,965,702	-
Chr 3	41.709.497	intron	1to5KB	GCBC	ULK4	41,246,599	41,962,103	-
Chr 3	187.931.631	intergenic	1to5KB	GCBC	AC068295.1	187,932,764	187,946,221	-
Chr 4	108.095.668	intron	promoter	GCBC	LEF1	108,047,548	108,168,932	-
Chr 5	1.285.859	intron	1to5KB	GCBC	TERT	1,253,167	1,295,068	-
Chr 5	132.662.721	intron	promoter	MBC	TH2LCRR	132,638,076	132,664,272	-
Chr 5	175.054.011	intergenic	1to5KB	CLL	LINC01951	174,919,082	174,995,513	-
Chr 6	33.716.536	TF_binding_site	promoter	MCL	UQCC2	33,696,764	33,711,700	-
Chr 7	50.402.283 – 50.405.553	3_prime_UTR	1to5KB	NBC	IKZF1	50,304,716	50,405,101	+
Chr 7	50.409.816	intergenic	1to5KB	NBC	IKZF1	50,304,716	50,405,101	+
Chr 8	127.210.176	intron	1to5KB	MBC	CASC19 / PCAT1	127,078,895/ 126,865,908	127,185,243/ 127,064,901	-/+
Chr 8	128.064.327 - 128.180.025	intron	1to5KB	PBC	PVT1	127,794,559	128,101,254	+
Chr 9	21.976.403	intron	promoter	CLL	CDKN2A / AL359922.1	21,967,752/ 21,802,636	21,994,392/ 22,029,594	-/+
Chr 9	22.006.274	intron	promoter	GCBC	CDKN2B-AS1 / CDKN2B / AL359922.1	21,995,133/ 22,002,903/ 21,802,636	22,128,103/ 22,009,305/ 22,029,594	+/-/+
Chr 10	22.557.806	intron	1to5KB	MCL	PIP4K2A	22,534,854	22,714,578	-
Chr 12	116.928.726	intron	1to5KB	GCBC	FBXW8	116,910,950	117,031,148	+
Chr 14	52.274.253	intron	promoter	MBC	PTGDR	52,267,698	52,276,724	+
Chr 14	92.231.568	intergenic	1to5KB	PBC	AL133240.1	92,253,493	92,263,656	-
Chr 14	103.018.488	intron	1to5KB	GCBC	CDC42BPB	102,932,380	103,057,549	-
Chr 15	69.726.651 - 69.728.186	intron	promoter	MCL	DRAIC	69,628,784	69,840,828	+
Chr 16	85.922.065	3_prime_UTR	1to5KB	NBC	IRF8	85,899,162	85,922,606	+
Chr 18	63.121.512	intergenic	1to5KB	GCBC	BCL2	63,123,346	63,320,128	-
Chr 18	63.126.261 - 63.126.688	3_prime_UTR	1to5KB	GCBC	BCL2	63,123,346	63,320,128	-
Chr 19	1.650.135	intron	1to5KB	NBC	TCF3	1,609,292	1,652,615	-
Chr 22	39.150.140	intron	promoter	NBC	CBX7	39,130,772	39,152,680	-
Chr 22	50.532.837	reg_region	promoter	CLL	ODF3B	50,530,409	50,532,579	-

Tabla 2: En esta tabla se recogen todas las variantes seleccionadas en los pasos anteriores, es decir aquellas que influyen en la región promotora o cercana a ésta (1to5KB). En ella se indican el cromosoma en el que se presenta la variante, su coordenada en dicho cromosoma, dónde se encuentra la variante, la región complementaria en la que ejerce influencia, la línea celular en la que se encuentra la región complementaria donde influye, el gen en el que se encuentra la variante (aquellos que aparecen en fondo blanco indica que es un gen cercano, pero que la variante no se encuentra dentro de él), las coordenadas de inicio y fin del gen dentro del cromosoma indicado en la primera columna, y por último la cadena en la que se encuentra el gen (+ o -).

Tras obtener esta información, fuimos capaces de realizar un análisis más exhaustivo, ya que ahora podíamos saber si la variante que afecta a la transcripción de un gen al otro lado del bucle, se encontraba dentro de éste (el gen tiene un tamaño similar al del bucle o termina aproximadamente donde el SNP afecta), o en una zona cercana.

La primera familia de genes que decidimos analizar fue la de *CDKN2*, ya que observamos un comportamiento interesante en estos genes. Por un lado, un SNP en un intrón dentro del cromosoma 9, parece afectar al promotor de los genes *CDKN2B* y *CDKN2B-AS1* en la línea celular GCBC. Mientras, cuando las células se transforman en células de LLC, observamos una variante cercana en el cromosoma 9. Esta variante se encuentra también en un intrón, pero que esta vez se sitúa en el interior del gen *CDKN2A*, muy cercanos en el espacio a los genes *CDKN2B* y *CDKN2B-AS1*. Estos inhibidores de la quinasa dependiente de ciclina 2A se que se encargan de controlar la proliferación celular, siendo conocidos por actuar como supresores de tumores [56] [57]; por lo tanto, parece que mutaciones como estas, que afectase al promotor de estos genes, por ejemplo, eliminando un binding site para un factor de transcripción, sumada a otras mutaciones detectadas en el interior de estos genes, especialmente del *CDKN2A* (como se puede observar en la tabla del Anexo), Este fenómeno podría darse, en una primera instancia en células de tipo GCBC afectando al gen *CDKN2B*, y posteriormente en LLC, apareciendo nuevas mutaciones afectando al gen *CDKN2A* (tanto linealmente como mediante contactos de largo alcance en bucles como hemos estudiado). La combinación de estos factores podría provocar una disminución de la actividad supresora de tumores, y por ende, un aumento en actividad tumoral y en la probabilidad de padecer LLC, pudiendo tener también peores previsiones [Figura 18].

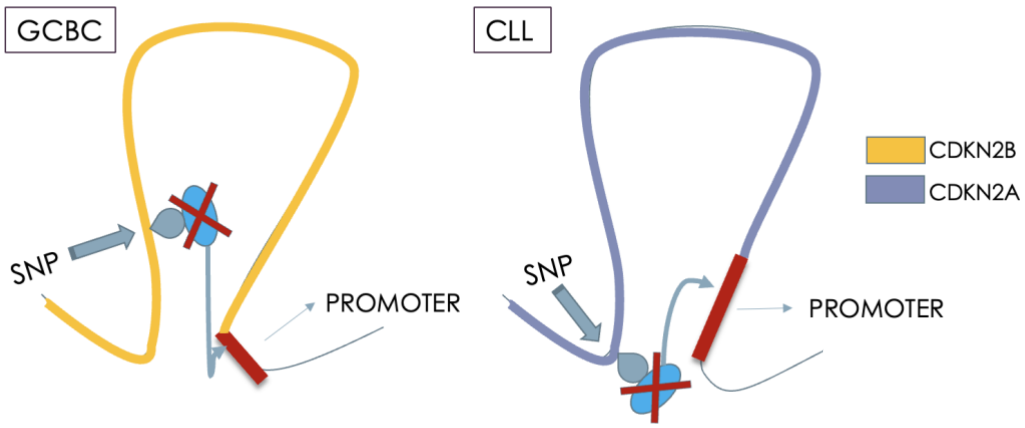


Figura 18: Representación gráfica de las interacciones entre los SNPs presentes en los intrones, tanto de GCBC en primer lugar, y de LLC después, y cómo estos pueden afectar al binding site de una proteína (representada en azul) que entraría en contacto con la región promotora de los genes *CDKN2B* (en el caso de GCBC) y *CDKN2A* (en el caso de LLC), pudiendo inhibir así su activación y funcionamiento normal codificando proteínas supresoras de tumores.

A continuación, analizamos el gen *IKZF1*. Un conjunto de mutaciones presentes en la zona 3'-UTR cercana al gen, parecían afectar a la zona de 1 a 5 kilobases del promotor de este

gen, en las células *Naive B-Cell (NBC)*. Además, otros dos SNPs presentes en la zona intergénica anterior a este gen, parecía influir en la misma zona debido a la formación de los bucles en la cromatina. De la misma manera que en el caso anterior, estas mutaciones pueden afectar a la unión del factor de transcripción, y por tanto, a la correcta activación del gen *IKZF1*. En concreto, este gen se encarga de la codificación de la proteína Ikaros, una proteína de unión al ADN y que funciona como reguladora del desarrollo de las células inmunes, principalmente en las células-B tempranas (lo cual coincide con haber encontrado esta interacción en la línea NBC, una de las primeras fases de las células-B). Además, se ha estudiado que la disminución o supresión de esta proteína, la cual se ha descubierto en los últimos años que es un importante supresor tumoral, está relacionada con el desarrollo de diversos tipos de leucemia, entre los que se encuentran la LLA (leucemia linfocítica aguda) [58] o la LLC [59], nuestro caso de estudio. Por lo tanto, en el caso de que las variantes surgidas en las zonas próximas al gen *IKZF1* afectasen a su transcripción debido a los bucles de la cromatina, como parece suceder según nuestro estudio, y la codificación de la proteína fuese totalmente suprimida o disminuyese, esto tendría implicaciones directas en el desarrollo de LLC. Este variación podría influir ya desde una etapa temprana, cuando las células aún estuviesen en la fase NBC, pudiendo ser ya un marcador de esta enfermedad desde edades muy tempranas [Figura 19].

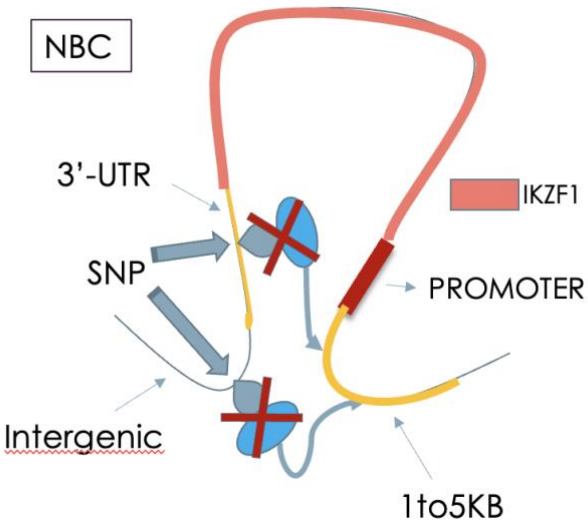


Figura 19: Modelo de interacciones entre las variantes encontradas en las proximidades del gen *IKZF1* (3'-UTR y la zona intergénica justo anterior), y su influencia en la zona cercana al promotor de este gen (1to5KB), pudiendo evitar la correcta aparición de la proteína codificada por este gen.

De manera similar al modelo representado del gen IKZF1, sucede con el gen BCL2, existiendo también cuatro mutaciones que afectan a la transcripción de este gen, una en una zona intergénica y otras tres en la zona 3'-UTR, que también influyen en la transcripción de este gen entrando en contacto con la zona 1to5KB al formarse el bucle. La diferencia con el gen anterior, además de que esta interacción la encontramos en la línea celular GCBC en vez de en NBC, es que en este caso, parece ser que los SNPs crean nuevos binding sites,

aumentando así la expresión del gen BCL2. Avalado por la literatura [60], un nivel más alto de la proteína codificada por este gen, parece estar relacionada con la patogénesis y la progresión de LLC, siendo una característica adversa una expresión mayor de este gen.

El siguiente gen analizado en este proceso, fue el gen *LEF1*, que fue uno de los primeros genes detectados, presente en el cuarto cromosoma, y la primera variante detectada en la zona de un intrón. Tras investigar sobre la relación de este gen con la progresión del cáncer, y más concretamente, en la progresión de LLC, descubrimos que este gen se encuentra altamente sobreexpresado en la leucemia linfocítica crónica, además de estar asociado con la progresión y el mal pronóstico de esta enfermedad, especialmente en fases tempranas de la formación de las células B de LLC, ya que se ha visto que suprimiendo este gen, la supervivencia de las células se ve reducida (pudiendo ser además un potencial *target* farmacológico bastante prometedor, de medicamentos como el ácido etacrínico [61] [62]). También se encuentra asociado con la progresión de otras enfermedades similares, como podría ser el cáncer colorrectal. Por este motivo, deducimos que los SNPs que afectan a este gen, actúan creando nuevos lugares de unión para proteínas que favorezcan la activación del promotor del *LEF1*, aumentando su transcripción, sobreexpresándolo y contribuyendo a la evolución de LLC, ya desde fases tempranas, cuando las células aún son de tipo GCBC [Figura 20].

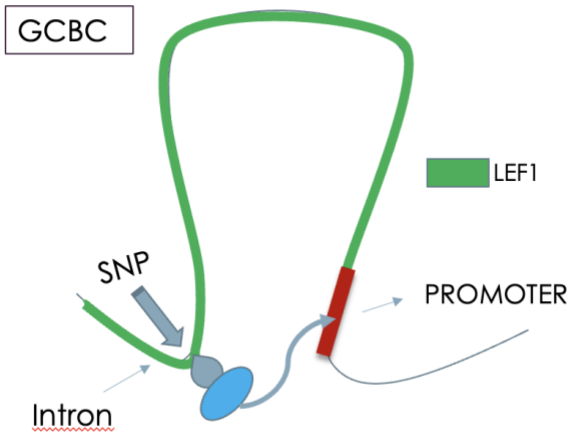


Figura 20: Representación gráfica de la interacción generada por la formación de un bucle entre el SNP presente en un intrón en el cuarto cromosoma y el promotor del gen *LEF1*, cuya alteración puede aumentar la transcripción de este gen a través de la unión de un mayor número de proteínas.

La última interacción variante-gen a través de un bucle que analizaremos es la formada en el segundo cromosoma. En este caso, un SNP mapeado en las células B plasmáticas, en un intrón, es probable que contribuya a la sobreexpresión del gen *SP3*, creando un nuevo *binding site* para una proteína, que a su vez entra en contacto con la zona próxima al promotor de este gen al crearse el bucle (concretamente, como en casos anteriores, en la zona situada entre 1 y 5 kilobases de éste), favoreciendo un aumento en la transcripción del gen *SP3* [Figura 22]. Estudiando la influencia de este gen en la progresión cancerosa, hemos

descubierto que, a pesar de no encontrarse tan relacionado directamente con la evolución de la leucemia como en los anteriores casos, si que aparece desregulado en muchos tipos de cánceres [63], como el de mama [64] o el de próstata [65], por lo que probablemente también juegue algún papel similar en los diferentes tipos de leucemia aunque no haya sido estudiado aún en profundidad.

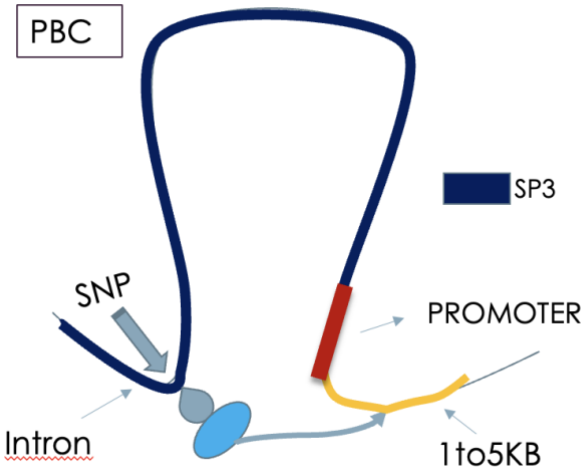


Figura 21: Representación gráfica de la interacción generada por la formación de un bucle entre el SNP presente en un intrón en el segundo cromosoma y la zona comprendida entre 1 y 5 kilobases del promotor del gen *SP3*, cuya alteración puede aumentar la transcripción de este gen.

Aparte de los genes representados gráficamente en este apartado, hubo otros que destacamos en el análisis de enriquecimiento, y que posteriormente también investigamos, y además de tener un comportamiento similar a los estudiados en estos ejemplos, tenían algún tipo de influencia en la progresión de diversos tipos de cáncer, entre ellos algunos directamente con la LLC.

Algunos de ellos fueron, por ejemplo, el gen *TERT*, que además de formar parte de la unidad más importante del complejo de la telomerasa, juega un rol fundamental en el número de veces que una célula es capaz de dividirse, teniendo así un importante papel en la inmortalidad de las líneas celulares, y dependiendo de su inhibición, o como resultado de la disminución de la proteína codificada por este gen, puede contribuir a la formación de células cancerosas inmortales. Es un gen que ha sido estudiado en el desarrollo de varios cánceres [66].

Otro gen que parece estar relacionado con el desarrollo de las células B en la leucemia es el gen *TCF3* [67], ya que se ha estudiado que la disminución de su actividad o la inhibición de su transcripción, parece estar relacionada con la aparición de malignidades en los linfocitos prematuramente, especialmente en niños y jóvenes, lo cual coincide con haber encontrado que esta mutación afectaba desde un comienzo a las células NBC, una de las primeras fases de las células B, pudiendo estar presente esta variante desde el nacimiento o desde edades

muy tempranas, permitiendo su estudio como marcador del desarrollo de malignidades relacionadas con las células B.

La transcripción de estos dos genes parecía verse afectada por SNPs presentes en zonas intrónicas, ayudadas por la formación de bucles. Sin embargo, en el caso del gen *IRF8*, este SNP se encuentra en la zona 3'-UTR de dicho gen, que al igual que en los casos anteriores, parece que es potencialmente un gen encargado de la supresión tumoral, y cuya inhibición podría contribuir a la formación y evolución de tumores [68], y al igual que en el caso anterior, al haberse encontrado esta interacción presente en la línea celular NBC, esta mutación podría desembocar en la aparición de LLC desde tiempo antes de que las células evolucionen [69].

Existe otro conjunto de genes presentes en el análisis de enriquecimiento para LLC o enfermedades relacionadas, pero no han sido comentados aquí, o bien por no haberse encontrado evidencias o estudios que corroboren su influencia en la aparición o desarrollo del cáncer, o mas concretamente, de LLC, o bien porque no aparecían enriquecidos para las enfermedades, ontologías o *pathways* que estábamos buscando (como podrían ser los genes *CASC19* o *ODF3B*).

A continuación de todo este procedimiento, el siguiente paso para saber como los modelos propuestos previamente afectarían realmente a la transcripción genética y como podrían influir en la progresión de LLC, sería investigar el efecto de la variante en la cadena de ADN.

Un ejemplo de cómo podríamos hacer esto, lo hemos realizado con la variante que se encuentra en un intron del gen *CDKN2B.* Para ello, consultamos nuestro fichero de variantes con la coordenada del SNP seleccionado, y obtuvimos el ID de referencia (*rs*). Con el identificador, buscamos en el GWAS Catalog la variante, donde pudimos observar cual era el cambio de nucleótido que provocaba (en este caso la base original era 'G', y la resultante tras el SNP, 'A').

Después, buscamos la coordenada de nuevo en el *UCSC Genome Browser*, y cogimos las diez bases alrededor de esta posición ('CTTTGGATAG'). En este caso nos ha servido con diez, pero en otros ejemplos podría ser necesario utilizar un numero mayor. Apuntamos también cual sería la secuencia de nucleótidos formada una vez realizado el polimorfismo ('CTTTGAATAG'), para estudiar que diferencias tendría en la transcripción.

Posteriormente, utilizando la herramienta *footPrintDB* [70], introdujimos las dos secuencias, la original y la mutada. Ordenamos los resultados en base a su relevancia (*e-value*), y comprobamos que, en el caso de la original, esta secuencia no cumpliría ninguna función relevante para nuestro estudio, pudiendo tener diversas funciones, ninguna totalmente determinada.

Sin embargo, al observar la secuencia mutada, comprobamos que el resultado con una mayor relevancia, era que actuaba como *binding site* para el factor de transcripción *SOX-9*.

Pudimos concluir que el efecto de este SNP no era exactamente el que habíamos asumido en el modelo que definimos previamente. En cambio, al formarse este SNP, se crea una zona unión para este factor de transcripción, que tras investigar sobre él [71], se trataba de un

gen con funciones oncogénicas, implicado en la tumorgénesis y sobreexpresado en varios tipos de cánceres diferentes.

Por tanto, la aparición de una zona de unión para este factor de transcripción encargado de expresar al gen *SOX-9*, parece estar ligado directamente con la progresión de LLC.

De la misma manera que hemos realizado con este SNP, y al no haber podido proseguir analizando otros por falta de tiempo, proponemos que se realizase un análisis similar con variantes similares, como las ya vistas en los modelos diseñados previamente, para identificar el efecto real de estas variantes en la transcripción de genes a los que podrían afectar a través de los bucles, y que podrían tener influencia en LLC o en otros tipos de cánceres o leucemias.

Conclusiones

A lo largo de este trabajo, hemos desarrollado una metodología que nos permite ver en cuatro dimensiones el efecto de los SNPS, en este caso en la leucemia linfocítica crónica.

Por un lado, en el espacio, estudiando como los SNPs pueden afectar a la transcripción de genes que se encuentran en zonas del genoma a priori alejadas de las variantes, pero que una vez se forman los bucles o pliegues en la cromatina, pasan a entrar en contacto, pudiendo tener influencia en las zonas promotoras o cercanas a éstas de los genes.

Por otro lado, en el tiempo, ya que a lo largo del proyecto hemos trabajado con seis líneas celulares diferentes, partiendo de las *Naive B-Cell*, que son uno de los estados iniciales de las células B, hasta llegar a la línea celular de LLC. De esta manera, hemos estudiado varias de las fases (NBC, GCBC, MBC, PBC, y además MCL, otro tipo de linfoma, y LLC ó CLL) que experimentan estas células hasta que pasan a ser de LLC, analizando de esta manera qué efecto tienen los SNPs en las distintas fases de la diferenciación de las células B, y como pueden afectar las variantes en etapas tempranas a la formación final de esta enfermedad.

La hipótesis de partida que teníamos era que un SNP que aparece en estados previos a LLC, puede dejar una huella o marca en las células que, mas adelante, cuando ya está la enfermedad, puede terminar produciendo estabilidades o inestabilidades en otros procesos que tienen impacto mas adelante, como por ejemplo a nivel epigenético, en el desarrollo y evolución de la enfermedad, así como en su diagnóstico, pudiendo causar que este sea más o menos favorable, e incluso modificar las alternativas de tratamiento.

Finalmente, aunque no hayamos alcanzado resultados totalmente definitivos, este análisis nos permite establecer un punto de estudio a través de los diferentes modelos de posibles mecanismos que pueden derivar en leucemia linfocítica crónica, ya sea por inhibición de la transcripción de determinados genes (como los genes de la familia CDKN2, o los genes IKZF1 y BCL2) o por sobreexpresión de otros (como los genes LEF1 o SP3), atendiendo a varias etapas de la diferenciación de las células B.

A partir de este estudio, proponemos aplicar este análisis siempre que sea posible a otros conjuntos de datos de otros tipos de cáncer o enfermedades diferentes, ya que como hemos comprobado a lo largo del proyecto, las fases previas al desarrollo del propio cáncer son igualmente importantes, y estas variaciones en etapas tempranas de su desarrollo pueden terminar afectando a su pronóstico.

En adición, también sería interesante realizar en un futuro un análisis exhaustivo y en profundidad del efecto de los SNPs en las zonas de unión de los factores de transcripción, para comprobar las consecuencias reales de las variantes en la expresión de los diferentes genes afectados por ellas al formarse los bucles en la cromatina, y cómo esto influye en la evolución de las enfermedades estudiadas, como es LLC en nuestro caso, de la misma manera en la que nosotros hemos analizado el caso de la variante presente en el gen CDKN2B. Otra posible ampliación sería trabajar con otras resoluciones del genoma.

Anexo

GEN	CHR	COORD. INICIO	COORD. FIN	SNPS
SP3	2	173,900,773	173,965,702	173944171
ULK4	3	41,246,599	41,962,103	41709497 41744517 41950916 41954644
AC068295.1	3	187,932,764	187,946,221	-3kb
LEF1	4	108,047,548	108,168,932	108095668 108104709
TERT	5	1,253,167	1,295,068	1279675 1285859
TH2LCRR	5	132,638,076	132,664,272	132662721
LINC01951	5	174,919,082	174,995,513	+6kb
UQCC2	6	33,696,764	33,711,700	+5kb
IKZF1	7	50,304,716	50,405,101	50398606 50402283 50402906
PCAT1	8	126,865,908	127,064,901	+120kb
CASC19	8	127,078,895	127,185,243	127183089
PVT1	8	127,794,559	128,101,254	128064327
CDKN2A	9	21,967,752	21,994,392	21970917 21976403 21984662 21991924
AL359922.1	9	21,802,636	22,029,594	21970917
CDKN2B-AS1	9	21,995,133	22,128,103	22006274
CDKN2B	9	22,002,903	22,009,305	22006274
PIP4K2A	10	22,534,854	22,714,578	22550699 22557806 22568017
FBXW8	12	116,910,950	117,031,148	116928726
PTGDR	14	52,267,698	52,276,724	52274253
AL133240_1	14	92,253,493	92,263,656	-22kb
CDC42BPB	14	102,932,380	103,057,549	103018488
DRAIC	15	69,628,784	69,840,828	69697166 69726651 69728186
IRF8	16	85,899,162	85,922,606	85894208 85895015 85910833 85922065
BCL2	18	63,123,346	63,320,128	63126261 63126316 63126688
TCF3	19	1,609,292	1,652,615	1650135
CBX7	22	39,130,772	39,152,680	39146287 39150140
ODF3B	22	50,530,409	50,532,579	+0.3kb

Tabla 3: Genes analizados en el estudio y los SNPs registrados de CLL que caen en ellos. Se indica el nombre del gen, el cromosoma en el que se encuentra, las coordenadas de inicio y fin y los SNPs que caen dentro de ellos. Los que no tienen ninguna variante en su interior, se indica, en KB (y con + o -) cual sería el SNP más cercano

Bibliografía

[1] T. A. C. S. m. a. e. c. team, «What is Chronic Lymphocytic Leukemia?,» The American Cancer Society, 10 Mayo 2018. [En línea]. Available: https://www.cancer.org/cancer/chronic-lymphocytic-leukemia/about/what-is-cll.html. [Último acceso: 6 Abril 2021].

[2] Rozovski U, Keating MJ, Estrov Z. Why Is the Immunoglobulin Heavy Chain Gene Mutation Status a Prognostic Indicator in Chronic Lymphocytic Leukemia? *Acta Haematol*. 2018;140(1):51-54. doi: 10.1159/000491382.

[3] Slager et al. Medical History, Lifestyle, Family History, and Occupational Risk Factors for Chronic Lymphocytic Leukemia/Small Lymphocytic Lymphoma: The InterLymph Non-Hodgkin Lymphoma Subtypes Project, *JNCI Monographs*, Volume 2014, Issue 48, August 2014, Pages 41–51, doi.org/10.1093/jncimonographs/lgu001

[4] T. A. C. S. m. a. e. team, «What Are the Risk Factors for Chronic Lymphocytic Leukemia?,» The American Cancer Society, 10 Mayo 2018. [En línea]. Available: https://www.cancer.org/cancer/chronic-lymphocytic-leukemia/causes-risks-prevention/risk-factors.html. [Último acceso: 6 Abril 2021].

[5] Hallek et al. iwCLL guidelines for diagnosis, indications for treatment, response assessment, and supportive management of CLL. *Blood* 2018; 131 (25): 2745–2760. doi: doi.org/10.1182/blood-2017-09-806398

[6] Gaidano, G., & Rossi, D. The mutational landscape of chronic lymphocytic leukemia and its impact on prognosis and treatment. *Hematology. American Society of Hematology. Education Program*, *2017*(1), 329–337. https://doi.org/10.1182/asheducation-2017.1.329

[7] Visscher et al. Five Years of GWAS Discovery. *American Journal of Human Genetics,* 2012; 90(1): 7-24. doi.org/10.1016/j.ajhg.2011.11.029

[8] Kazuyuki Matsuda. PCR-Based Detection Methods for Single-Nucleotide Polymorphism or Mutation: Real-Time PCR and Its Substantial Contribution Toward Technological Refinement**.** *Advances in Clinical Chemistry,* 2017; 80: 45-72. doi.org/10.1016/bs.acc.2016.11.002

[9] MacArthur et al. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog), *Nucleic Acids Research*, Volume 45, Issue D1, January 2017, Pages D896–D901, https://doi.org/10.1093/nar/gkw1133

[10] Buniello et al, The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019, *Nucleic Acids Research*, Volume 47, Issue D1, 08 January 2019, Pages D1005–D1012, https://doi.org/10.1093/nar/gky1120

[11] Bauer-Mehren A., Rautschka M., Sanz F., Furlong L.I. DisGeNET: a Cytoscape plugin to visualize, integrate, search and analyze gene-disease networks. Bioinformatics. 2010; 26:2924–2926.

[12] Smigielski EM, Sirotkin K, Ward M, Sherry ST. dbSNP: a database of single nucleotide polymorphisms. Nucleic Acids Res. 2000 Jan 1;28(1):352-5. doi: 10.1093/nar/28.1.352

[13] Piñero et al, The DisGeNET knowledge platform for disease genomics: 2019 update, *Nucleic Acids Research*, Volume 48, Issue D1, 08 January 2020, Pages D845–D855, https://doi.org/10.1093/nar/gkz1021

[14] Gallagher, M. D., & Chen-Plotkin, A. S. (2018). The Post-GWAS Era: From Association to Function. *American journal of human genetics*, *102*(5), 717–730. https://doi.org/10.1016/j.ajhg.2018.04.002

[15] Tak, Y.G., Farnham, P.J. Making sense of GWAS: using epigenomics and genome engineering to understand the functional relevance of SNPs in non-coding regions of the human genome. *Epigenetics & Chromatin* 8, 57 (2015). https://doi.org/10.1186/s13072-015-0050-4

[16] Carter D, Chakalova L, Osborne CS, Dai YF, Fraser P. Long-range chromatin regulatory interactions in vivo. *Nat Genet.* 2002;32:623–6

[17] Kurukuti et al. CTCF binding at the H19 imprinting control region mediates maternally inherited higher-order chromatin conformation to restrict enhancer access to Igf2. *Proc Natl Acad Sci U S A.* 2006;103:10684–9

[18] Kadauke, S., & Blobel, G. A. (2009). Chromatin loops in gene regulation. *Biochimica et biophysica acta*, *1789*(1), 17–25. https://doi.org/10.1016/j.bbagrm.2008.07.002

[19] Lieberman-Aiden E, van Berkum NL, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science.* (2009), 326(5950):289-93

[20] Belton, J. M., McCord, R. P., Gibcus, J. H., Naumova, N., Zhan, Y., & Dekker, J. (2012). Hi–C: a comprehensive technique to capture the conformation of genomes. *Methods*, *58*(3), 268-276.
[21] Sanger, F., Nicklen, S., Coulson, A.R., (1977). DNA sequencing with chainterminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America* 74, 5463–5467.

[22] Behjati S, Tarpey PS What is next generation sequencing? *Archives of Disease in Childhood - Education and Practice* 2013; 98:236-238.

[23] Fakruddin, M., Chowdhury, A., Hossain, N., Mahajan, S., & Islam, S. (2013). Pyrosequencing: A next generation sequencing technology. *World Appl Sci J*, *24*(12), 1558-1571.

[24] Datto, M., & Lundblad, R. L. (2016). DNA, RNA chemical properties (including sequencing and next-generation sequencing). 24-35

[25] McCombie, W. R., McPherson, J. D., & Mardis, E. R. (2019). Next-generation sequencing technologies. *Cold Spring Harbor perspectives in medicine*, *9*(11), a036798.

[26] Goodwin, S., McPherson, J. D., & McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, *17*(6), 333.

[27] Greenwald, W.W., Li, H., Benaglio, P. *et al.* Subtle changes in chromatin loop contact propensity are associated with differential gene regulation and expression. *Nat Commun* 10, 1054 (2019). https://doi.org/10.1038/s41467-019-08940-5

[28] Pal, K., Forcato, M. & Ferrari, F. Hi-C analysis: from data generation to integration. *Biophys Rev* 11, 67–78 (2019). https://doi.org/10.1007/s12551-018-0489-1

[29] Forcato, M., Nicoletti, C., Pal, K. *et al.* Comparison of computational methods for Hi-C data analysis. *Nat Methods* 14**,** 679–685 (2017). https://doi.org/10.1038/nmeth.4325

[30] Ay F, Bailey TL, Noble WS. Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. Genome Res. 2014; 24(6):999–1011.

[31] Carty M, Zamparo L, Sahin M, González A, Pelossof R, Elemento O, Leslie CS. An integrated model for detecting significant chromatin interactions from high-resolution hi-c data. Nat Commun. 2017; 8:15454.

[32] Rao SSP et al. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell. 2014; 159(7):1665–80

[33] Rowley MJ et al. Analysis of Hi-C data using SIP effectively identifies loops in organisms from C. elegans to mammals. Genome Res. 2020; 30(3):447–58

[34] Roayaei Ardakany, A., Gezer, H.T., Lonardi, S. *et al.* Mustache: multi-scale detection of chromatin loops from Hi-C and Micro-C maps using scale-space representation. *Genome Biol* 21, 256 (2020).

[35] Salameh, T.J., Wang, X., Song, F. *et al.* A supervised learning framework for chromatin loop detection in genome-wide contact maps. *Nat Commun* 11**,** 3428 (2020)

[36] Fullwood, M. J. et al. An oestrogen-receptor-alpha-bound human chromatin interactome. *Nature* 462, 58–64 (2009)

[37] Mifsud, B. et al. Mapping long-range promoter contacts in human cells with high-resolution capture Hi-C. *Nat. Genet.* **47**, 598–606 (2015)

[38] Qi Y. (2012) Random Forest for Bioinformatics. In: Zhang C., Ma Y. (eds) Ensemble Machine Learning. Springer, Boston, MA

[39] Joachim Wolff, Rolf Backofen, Björn Grüning. Loop detection using Hi-C data with HiCExplorer. *bioRxiv (preprinted)* (2020)

[40] Melissa J Landrum et al, ClinVar: improving access to variant interpretations and supporting evidence, *Nucleic Acids Research*, Volume 46, Issue D1, 4 January 2018, Pages D1062–D1067

[41] Sherry, S. T., Ward, M. H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., & Sirotkin, K. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic acids research*, *29*(1), 308-311.

[42] Speedy, H.E., Beekman, R., Chapaprieta, V. *et al.* Insight into genetic predisposition to chronic lymphocytic leukemia from integrative epigenomics. *Nat Commun* 10**,** 3615 (2019).

[43] Di Filippo L, Righelli D, Gagliardi M, Matarazzo MR and Angelini C (2019) HiCeekR: A Novel Shiny App for Hi-C Data Analysis. *Front. Genet.* 10:1079. doi: 10.3389/fgene.2019.01079

[44] Servant, N., Lajoie, B. R., Nora, E. P., Giorgetti, L., Chen, C. J., Heard, E., ... & Barillot, E. (2012). HiTC: exploration of high-throughput 'C' experiments. *Bioinformatics*, *28*(21), 2843-2844.

[45] Calandrelli, R., Wu, Q., Guan, J., & Zhong, S. (2018). GITAR: an open source tool for analysis and visualization of Hi-C data. *Genomics, proteomics & bioinformatics*, *16*(5), 365-372.

[46] Ron, G., Globerson, Y., Moran, D. *et al.* Promoter-enhancer interactions identified from Hi-C data using probabilistic models and hierarchical topological domains. *Nat Commun* 8**,** 2237 (2017)

[47] Xu Z, Zhang G, Wu C, Li Y, Hu M. FastHiC: a fast and accurate algorithm to detect long-range chromosomal interactions from Hi-C data. *Bioinformatics*. 2016;32(17):2692-2695.

[48] Wingett S, Ewels P, Furlan-Magaril M, et al. HiCUP: pipeline for mapping and processing Hi-C data. *F1000Res*. 2015;4:1310. Published 2015 Nov 20.

[49] Duttke, S. H., Chang, M. W., Heinz, S., & Benner, C. (2019). Identification and dynamic quantification of regulatory elements using total RNA. *Genome research*, *29*(11), 1836-1846

[50] Bhattacharyya, S., Chandra, V., Vijayanand, P. *et al.* Identification of significant chromatin contacts from HiChIP data by FitHiChIP. *Nat Commun* 10**,** 4221 (2019).

[51] Fernández, J. M., de la Torre, V., Richardson, D., Royo, R., Puiggròs, M., Moncunill, V., ... & BLUEPRINT Consortium. (2016). The BLUEPRINT data analysis portal. *Cell systems*, *3*(5), 491-495.

[52] Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. *PloS one*, *12*(5), e0177459.

[53] Serra, F., Baù, D., Goodstadt, M., Castillo, D., Filion, G. J., & Marti-Renom, M. A. (2017). Automatic analysis and 3D-modelling of Hi-C data using TADbit reveals structural features of the fly chromatin colors. *PLoS computational biology*, *13*(7)

[54] Durand, N. C., Shamim, M. S., Machol, I., Rao, S. S., Huntley, M. H., Lander, E. S., & Aiden, E. L. (2016). Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. *Cell systems*, *3*(1), 95-98.

[55] Abdennur, N., & Mirny, L. A. (2020). Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, *36*(1), 311-316.

[56] Xia, Y., Liu, Y., Yang, C. *et al*. Dominant role of *CDKN2B/p15INK4B* of 9p21.3 tumor suppressor hub in inhibition of cell-cycle and glycolysis. *Nat Commun* 12**,** 2047 (2021).

[57] Ran Zhao Bu YoungChoi, *et al*. Implications of Genetic and Epigenetic Alterations of CDKN2A (p16INK4a) in Cancer. *EBioMedicine* 8, 30-39 (2016).

[58] Vivian Cristina de Oliveira, Marcelo Pitombeira de Lacerda, *et al* (2019). Deregulation of Ikaros expression in B-1 cells: New insights in the malignant transformation to chronic lymphocytic leukemia. *Journal of Leukocyte Biology*, 106(3), 581-594.

[59] Payne KJ, Dovat S. Ikaros and tumor suppression in acute lymphoblastic leukemia. Crit Rev Oncog. 2011;16(1-2):3-12.

[60] Robertson LE, Plunkett W, McConnell K, Keating MJ, McDonnell TJ. Bcl-2 expression in chronic lymphocytic leukemia and its correlation with the induction of apoptosis and clinical outcome. Leukemia. 1996 Mar;10(3):456-9.

[61] Gutierrez A Jr, Tschumper RC, Wu X, et al. LEF-1 is a prosurvival factor in chronic lymphocytic leukemia and is expressed in the preleukemic state of monoclonal B-cell lymphocytosis. Blood. 2010;116(16):2975-2983.

[62] Wu W, Zhu H, Fu Y, et al. High LEF1 expression predicts adverse prognosis in chronic lymphocytic leukemia and may be targeted by ethacrynic acid. Oncotarget. 2016;7(16):21631-21643. doi:10.18632/oncotarget.7795

[63] Li, L., & Davie, J. R. (2010). The role of Sp1 and Sp3 in normal and cancer cell biology. Annals of Anatomy-Anatomischer Anzeiger, 192(5), 275-283.

[64] Mansour, M. A. (2021). SP3 is associated with migration, invasion, and Akt/PKB signalling in MDA-MB-231 breast cancer cells. Journal of biochemical and molecular toxicology, 35(3)

[65] Shin, T., Sumiyoshi, H., et al. (2005). Sp1 and Sp3 transcription factors upregulate the proximal promoter of the human prostate-specific antigen gene in prostate cancer cells. Archives of biochemistry and biophysics, 435(2), 291-302.

[66] Cao, Y., Bryan, T. M., & Reddel, R. R. (2008). Increased copy number of the TERT and TERC telomerase subunit genes in cancer cells. Cancer science, 99(6), 1092-1099.

[67] Ben-Ali, M., Yang, J., et al. (2017). Homozygous transcription factor 3 gene (TCF3) mutation is associated with severe hypogammaglobulinemia and B-cell acute lymphoblastic leukemia. Journal of Allergy and Clinical Immunology, 140(4), 1191-1194.

[68] Meyer, M. A., Baer, J. M., Knolhoff, B. L., Nywening, T. M., Panni, R. Z., Su, X., ... & DeNardo, D. G. (2018). Breast and pancreatic cancer interrupt IRF8-dependent dendritic cell development to overcome immune surveillance. Nature communications, 9(1), 1-19.

[69] Slager, S. L., Achenbach, S. J., Asmann, Y. W., Camp, N. J., Rabe, K. G., Goldin, L. R., ... & Cerhan, J. R. (2013). Mapping of the IRF8 gene identifies a 3′ UTR variant associated with risk of chronic lymphocytic leukemia but not other common non-Hodgkin lymphoma subtypes. Cancer Epidemiology and Prevention Biomarkers, 22(3), 461-466.

[70] Sebastian A, Contreras-Moreira B. footprintDB: a database of transcription factors with annotated cis elements and binding interfaces. Bioinformatics. 2014 Jan 15;30(2):258-65

[71] Aguilar-Medina M, Avendaño-Félix M, Lizárraga-Verdugo E, et al. SOX9 Stem-Cell Factor: Clinical and Functional Relevance in Cancer. *J Oncol*. 2019;2019:6754040. Published 2019 Apr 1.

[72] Kuleshov MV, Jones MR, Rouillard AD, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res. 2016;44(W1):W90-W97.

[73] De Vries, A., & De Vries, A. (2004). Elsevier's dictionary of symbols and imagery. Brill.

[74] Lu, Z., BRETONNEL COHEN, K., & Hunter, L. (2007). GeneRIF quality assurance as summary revision. In Biocomputing 2007 (pp. 269-280).

[75] Schonlau, M. (2002). The clustergram: A graph for visualizing hierarchical and nonhierarchical cluster analyses. The Stata Journal, 2(4), 391-402.

[75] Hunter, D. J., et al. (2007). A genome-wide association study identifies alleles in FGFR2 associated with risk of sporadic postmenopausal breast cancer. Nature genetics, 39(7), 870-874