UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Agronòmica i del Medi Natural

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Agronómica y del Medio Natural

Optimización, validación y aplicación a un caso real de una herramienta bioinformática para el análisis transcriptómico de organismos no modelo utilizando lecturas largas

Trabajo Fin de Grado

Grado en Biotecnología

AUTOR/A: Sobrino Sánchez, Isidro

Tutor/a: Forment Millet, José Javier

Cotutor/a externo: CONESA CEGARRA, ANA

Director/a Experimental: AMORIN DE HEGEDUS, ROCIO

CURSO ACADÉMICO: 2021/2022

Optimización, validación y aplicación a un caso real de una herramienta bioinformática para el análisis transcriptómico de organismos no modelo utilizando lecturas largas.

El desarrollo de las tecnologías de secuenciación de lectura larga como Pacific Bioscience (PacBio) o Oxford Nanopore han permitido un enorme avance en la secuenciación, no solo de genomas, sino también de ARN mensajero. La mayor longitud de las lecturas ha permitido el descubrimiento de transcritos noveles que las tecnologías de secuencia corta no habían sido capaces de detectar. Esto se debe principalmente a la capacidad de secuenciar la totalidad del transcrito del extremo 5' al 3', lo que permite prescindir del ensamblaje y reconstrucciones necesarias con las tecnologías de short-read. A pesar de la eficacia demostrada por los softwares de mapeo y ensamblaje utilizados, estos pueden generar artefactos y no resuelven correctamente loci complejos. Es por ello por lo que muchos transcritos no son detectados al mapear sus lecturas con otros transcritos similares. Evitar este paso es crucial para tener una visión más clara de la estructura y los eventos de splicing que ocurren en cada gen, motivo por el cual cada vez se utilizan más las lecturas largas para estudios de secuenciación transcriptómica.

Junto al desarrollo de toda esta tecnología se hizo necesaria la aparición de nuevas herramientas bioinformáticas que permitan el tratamiento y análisis de estos datos generados. En el caso concreto de los estudios transcriptómicos resaltan especialmente aquellos dedicados al análisis, clasificación e incluso descubrimiento de nuevas isoformas producto de eventos de splicing. El splicing alternativo es uno de los principales mecanismos productores de diversidad en los seres vivos, resultando muy importante en prácticamente todos los procesos biológicos. Sin embargo, la mayoría de estas herramientas necesitan el apoyo de un genoma de referencia para clasificar y analizar las secuencias generadas. Esto dificulta su uso en aquellas especies que o bien carecen de este o no tiene suficiente calidad, como son la mayoría de las especies no-modelo. En la actualidad existen algunas herramientas de software que abordan el problema del análisis de datos de transcriptómica de long reads sin la utilización de una anotación de referencia. Sin embargo, quedan por desarrollar métodos que describan adecuadamente los transcriptomas resultantes.

El objetivo de este trabajo de fin de grado es la optimización y validación de un pipeline capaz de clasificar transcritos procedentes de proyectos de secuenciación con long-reads que no usan una anotación de referencia en su análisis. Esta clasificación es doble: por un lado, agrupa aquellos transcritos procedentes de un mismo gen y por otro clasifica los mismos en función del evento de splicing que parece haber sufrido (retención intrónica, cambios exónicos, UTRs alternativas, etc). El pipeline, principalmente programado en Python, utiliza diversos métodos que incluyen el mapeo, agrupamiento o ensamblaje, utilizando desde k-mers hasta grafos *de Bruijn*. Además, la metodología desarrollada se aplica a un caso real: el análisis transcriptómico de la especie *Micropterus salmoides*, utilizando un proyecto de secuenciación de RNA realizado previamente con la tecnología de secuenciación SMRT (PacBio).

Palabras clave: lecturas-largas; transcriptómica; isoformas; bioinformatica; no-modelo

Optimisation, validation and application to a real case of a bioinformatics tool for transcriptomic analysis of non-model organisms using long-reads.

The development of long-read sequencing technologies such as Pacific Bioscience (PacBio) or Oxford Nanopore has enabled a huge advance in sequencing, not only of genomes, but also of messenger RNA. Longer read lengths have enabled the discovery of novel transcripts that short sequencing technologies had not been able to detect. This is mainly due to the ability to sequence the entire transcript from the 5' to the 3' end, which makes it possible to dispense with the assembly and reconstructions required with short-read technologies. Despite the efficiency demonstrated by the mapping and assembly software used, these can generate artefacts and do not correctly resolve complex loci. This is why many transcripts are not detected when mapping their reads to other similar transcripts. Avoiding this step is crucial to get a clearer picture of the structure and splicing events occurring in each gene, which is why long reads are increasingly used for transcriptomic sequencing studies.

Along with the development of all this technology, new bioinformatics tools were needed to process and analyse the data generated. In the specific case of transcriptomic studies, those dedicated to the analysis, classification and even discovery of new isoforms resulting from splicing events stand out. Alternative splicing is one of the main diversity-producing mechanisms in living beings and is very important in practically all biological processes. However, most of these tools need the support of a reference genome to classify and analyse the sequences generated. This makes it difficult to use them in species that either lack a reference genome or do not have one of sufficient quality.

The aim of this final degree project is the optimisation and validation of a pipeline capable of classifying transcripts from sequencing projects with long-reads. This classification is twofold: on the one hand, it groups transcripts from the same gene and, on the other, it classifies them according to the splicing event they appear to have undergone (intronic retention, exonic changes, alternative UTRs, etc.). The pipeline, mainly programmed in Python, uses a variety of methods including mapping, clustering or assembly, using everything from k-mers to bruijn graphs. In addition, it will be applied to a real case: transcriptomic analysis of the *Micropterus salmoides* species will be performed, using an RNA sequencing project previously carried out with SMRT sequencing technology (PacBio).

Key words: long-reads; transcriptomics,splicing,pipeline,non-model

Agradecimientos.

En primer lugar, agradecer a Ana el haberme dado la oportunidad de aprender y realizar este proyecto en su laboratorio. A Rocío, gracias por guiarme y ayudarme a pesar de lo difícil que los más de 7000 km que nos separaban lo ponían. Gracias por hacerme entender que un mal resultado también es un resultado, y que la ciencia es frustrante pero bonita.

A mis padres y hermanos, por enseñarme lo que es trabajar y ganarte tu sitio partiendo de 0. Gracias, mamá por enseñarme que ser buena persona es lo primero en esta vida, gracias Mima por enseñarme lo que es el amor incondicional a los demás y por dármelo cada vez que lo he necesitado.

A las bebés de mi casa, por llenarme la vida de alegría cada vez que os cojo en brazos.

Gracias a Mamen y Andreu por ser mi refugio y casa en estos 4 años, y sobre todo perdón por todas vuestras chuches que me he comido. Mamen, me has enseñado que la constancia y la fuerza de voluntad lo pueden absolutamente todo en esta vida, siempre vas a ser mi ejemplo a seguir.

Este es el resultado final de 4 años que habrían sido insoportables sin mis bioperras, gracias por tanta ruta, viaje y llanto. En especial quiero agradecer a Aitana y Ángela por haber estado ahí cuando poca gente lo hacía, espero y estoy seguro de que nuestros caminos no se van a separar.

Y por último muchísimas gracias a mi Carmen y a mi Luca, por enseñarme cada uno a vuestra manera que la familia también se elige, por recordarme lo que es ser querido todos los días y sobre todo por no hacerme dudar ni un segundo de que, pase lo que pase, nunca me voy a arrepentir de haberos dedicado estas palabras. Os amo con locura.

I. ÍNDICE

II. ÍNDICE DE FIGURAS

DEFINICIONES Y ABREVIATURAS:

Output: Archivos obtenidos a partir de utilizar un programa informático

Pipeline: Cadena de procesos informáticos conectados caracterizados por ser la entrada de uno la salida (output) del anterior

EF: fragmento exónico

RI: Retención intrónica

A5: Uso de dondador de splicing (5') alternativo

A3: Uso de aceptor de splicing (3') alternativo

EFE: exclusión de un fragmento exónico

EFI: Inclusión de un fragmento exónico

A5E: exclusión de un evento A5

A5I: Inclusión de un evento A5

A3I: Inclusión de un evento A3

A3E: Exclusión de un evento A3

BLT: Transcrito base (transcrito utilizado como referencia en un gen.

ESTs: Técnica consistente en identificar un transcrito secuenciando pequeños subfragmentos de este. La abreviatura viene del inglés: expressed sequence tag

RNA: Abreviatura de ácido desoxiribonucleico (ARN) en ingles

mRNA: Abreviatura de ARN mensajero en inglés.

GTF: Formato de archivo informático en el que encontramos información genómica sobre la estructura de los genes

FASTA: Formato de archivo informático en el que encontramos secuencias completas y el nombre de estas.

1. INTRODUCCIÓN.

Durante los últimos 20 años se ha vivido una revolución tecnológica que nos ha permitido desarrollar un nuevo tipo de ciencias. Estas se denominan las ciencias òmicas, que nos permiten la detección de forma universal de genes (genómica), ARN mensajero (ARNm) y otros tipos de ARN (transcriptómica), proteínas (proteómica) y metabolitos (metabólica) (Manzoni et al., 2018).

La transcriptómica se encarga del estudio del transcriptoma completo, lo cual implica el estudio de todos los ARNs que encontramos en una célula, desde microARN hasta los ARN largos no codificantes, pasando por los clásicos mensajeros (Lowe et al., 2017). La publicación del genoma humano en principios de los 2000 (Craig Venter et al., 2001; Lander et al., 2001) ha influido enormemente en esta ciencia. La transcriptómica ha experimentado un enorme aumento en el número de publicaciones desde que éste fuese publicado, aumentando aún más en los últimos años con la aparición de nuevos genomas completos de otras especies (Lowe et al., 2017).

Al principio, la mayoría de estas publicaciones se centraban sobre todo en el estudio de la expresión diferencial en diferentes condiciones o tejidos. Otro tipo de estudios se hacían más complicados debido a las limitaciones de los primeros métodos usados, que no eran capaces de revelar la secuencia completa de los transcritos. Estas técnicas basaban los experimentos en la secuenciación de pequeños fragmentos aleatorios del transcrito, como en la técnica denominada secuenciación de ESTs (Marra et al., 1998) o en la presencia de algunas secuencias clave en ellos, como en los muy utilizados microarrays (Schena et al., 1995). Sin embargo, tras el desarrollo de nuevas tecnologías como la secuenciación de corta longitud y protocolos específicos para la secuenciación de ARN (denominado RNA-seq) se abrió la puerta a nuevas posibilidades, debido a la mayor cantidad de información que se generaba (McGettigan, 2013).

El RNA-seq ha conseguido ampliar los estudios de expresión diferencial, pues permite la detección y cuantificación de genes con niveles de expresión baja (Sims et al., 2014). Además, se empezó a investigar la relevancia de los ARN no codificantes (T. Li et al., 2021; Pinkney et al., 2021), que constituyen una gran parte del transcriptoma y que tienen en su mayoría funciones reguladoras, así como los micro ARN (Benesova et al., 2021), un subtipo de los primeros con una longitud muy pequeña. Todos estos estudios han llevado a importantes descubrimientos en campos tan diversos como la respuesta a estreses en plantas (K. Zhang et al., 2022), el descubrimiento de biomarcadores en diferentes enfermedades (Jiang et al., 2021), así como los mecanismos patológicos de estas (Neff et al., 2021).

Además, gracias al paralelo avance de la bioinformática y las herramientas de mapeo de secuencias, se hizo posible no solo detectar partes de los transcritos, sino también la reconstrucción de sus secuencias, permitiendo así avanzar en otro aspecto de la transcriptómica: el estudio de los eventos de splicing (H. Feng et al., 2013).

1.1. La importancia del splicing.

Antes de la publicación de los resultados del proyecto genoma humano, basándose en la complejidad y diversidad que se conocía de nuestra especie, se pensaba que el genoma humano tendría hasta 150.000 genes, tal y como se muestra en la bibliografía científica del año 2000 (Pennisi, 2000). El resultado final esclareció que las estimaciones estaban muy lejos de la realidad: se demostró que el genoma humano contenía apenas 32.000 genes (Claverie, 2001); no mucho más que otras especies consideradas mucho menos complejas como *Caenorhabditis elegans* (19.000 genes) (Consortium*, 1998).

A partir de entonces, la búsqueda por entender la fuente de gran complejidad humana, que no eran los genes, llevó a abrir o revivir nuevos campos en la genética. Una de las causas de esta complejidad se encuentra en el transcriptoma: existen muchos más transcritos diferentes que genes, lo que eleva exponencialmente el número de proteínas que se pueden codificar a partir de la secuencia de ADN (Y. Liu et al., 2017). Estos diferentes transcritos se generan a partir de las mismas secuencias de ADN, gracias al mecanismo denominado *splicing alternativo (AS)*. Este mecanismo, inicialmente descubierto en adenoviruses (Berget et al., 1977), está presente en los organismos eucariotas y es posible gracias a la naturaleza de los genes en este tipo de células. Los genes eucariotas están compuestos por secuencias no codificantes, los intrones, los cuales están presentes en los estadíos iniciales de los transcritos: el pre-mRNA, y por exones, que si permanecen en el mRNA. Más adelante son eliminados por el spliceosoma, un complejo proteico que se ensambla sobre el mRNA. No se ensambla en cualquier lugar, lo hace sobre unos sitios con secuencias consenso a ambos lados de los intrones (denominados sitios aceptores y donadores de splicing) lo que lleva a su eliminación (Park et al., 2018).

También existen otros elementos auxiliares, adyacentes o cercanos a los sitios consenso que se encargan de "silenciar" o "potenciar" los sitios de splicing (Lei & Vořechovský, 2005). Esto provoca variaciones en el uso de estos sitios de splicing: algunos sitios pueden ser silenciados, por lo que no serán usados, mientras que secuencias similares presentes en los exones pueden ser confundidas como un sitio aceptor/donador si hay un potenciador cerca.

De esta forma, existen múltiples combinaciones que llevan a la diversidad transcriptómica: Algunos transcritos pueden tener algún intrón retenido o saltarse algún exón, entre otras posibilidades. Gracias a esto la variabilidad proteica se incrementa de forma espectacular, por ejemplo, los tres genes de las neuroxinas humanas son capaces de codificar un total de más de 2000 proteínas diferentes gracias al splicing alternativo (Tabuchi & Südhof, 2002). Además, el AS juega un papel muy relevante en importantes procesos biológicos, por ejemplo, los mecanismos que determinan si el desarrollo sexual es masculino o femenino en *Drosophila melanogaster* (mosca de la fruta, o también nombrada simplemente como Drosophila) viene determinado por el uso de sitios aceptores alternativos en el gen tra (Nagoshi & Baker, 1990). Los eventos de splicing también están detrás de los mecanismos patológicos de diferentes enfermedades como el cáncer (H. Feng et al., 2013; Y. Zhang, Qian, et al., 2021), estimándose que en torno al 50% de las variantes genéticas causantes de enfermedades en humanos afectan a algún evento de splicing haciéndolo aberrante (D. Feng & Xie, 2013) .

Desde el descubrimiento del splicing alternativo se han utilizado diferentes clasificaciones y nomenclaturas para denominar los diferentes tipos de eventos que podían tener lugar, sin embargo, hoy en día predominan 4 tipos de eventos en la bibliografía (Park et al., 2018). El primero de ellos se trata de la retención intrónica, evento que ocurre cuando un intrón no es eliminado y acaba formando parte del transcrito; el segundo es la exclusión de un exón (exon skipping en inglés) que ocurre cuando en un transcrito no está presente alguno de los exones del gen, asumiendo que siempre están presentes (Park et al., 2018). Los dos eventos restantes ocurren cuando en los extremos intrónicos existen varios sitios aceptores o donadores unos cerca de otros, separados por unas pocas bases. Estos sitios aceptores y donadores, configurados en tándem llevan a que aparezcan transcritos que difieren en unos pocos nucleótidos, pues cada uno ha utilizado uno de los distintos sitios de splicing disponibles, estos eventos se denominan uso de aceptor alternativo (o sitio 3' alternativo, A3 por sus siglas en inglés) y uso de donador alternativo (o sitio 5' alternativo, A5 por sus siglas en inglés) (Mayeda' And & Ohshima2, 1988). Sin embargo, también se han clasificado otros tipos de splicing menos comunes como la exclusión mutua de exones (mutually exclusive exons, ME) en la cual dos exones de un gen nunca aparecen juntos en sus transcritos o la elección de un primer exón diferente debido a el uso de diferentes promotores por la maquinaria de transcripción (Park et al., 2018). En la figura 1 encontramos ilustradas unas representaciones gráficas de cada uno de estos eventos.

En definitiva, por todo lo expuesto anteriormente, el estudio del splicing alternativo a nivel genómico juega un papel fundamental en el entendimiento de los organismos vivos, por lo que es fundamental desarrollar estrategias eficientes para su detección.

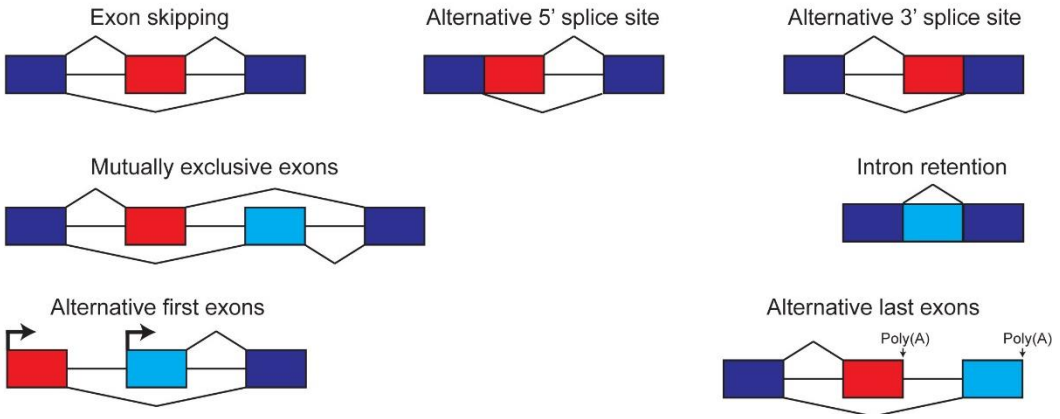

Figura 1. Clasificación de los eventos de splicing. Diagramas en los que se observan los principales eventos de splicing descritos en la bibliografía según su nombre en inglés. Retención intrónica (Intron retention), Exclusión de un exón (exon skipping), elección de un sitio 5' o 3' alternativo (alternative 5'/3' splice site), Exones mutuamente excluyentes (Mutually exclusive exons), y elección alternativa del último/primer exon (alternative last/first exon). Imagen de Park et al., 2018

1.2. Detección de Splicing alternativo.

A pesar de que los eventos de splicing se conocen desde 1977 (Berget et al., 1977) y que han sido investigados desde diferentes aproximaciones experimentales, no es hasta la llegada del RNA-seq cuando se empezó a estudiar exhaustivamente estos procesos (H. Feng et al., 2013) El RNA-seq proporciona información más o menos completa sobre la secuencia de los diferentes transcritos, lo que permite que mediante estrategias bioinformática podamos comparar las diferentes isoformas (transcritos de un mismo gen) y detectar así los eventos de splicing (Halperin et al., 2021). Sin embargo, la corta longitud de las lecturas obliga a realizar una reconstrucción bioinformática de los transcritos a partir de las lecturas obtenidas; esto implica la presencia de errores en las secuencias reconstruidas, impidiendo una detección óptima de los eventos.

Entre los problemas encontrados en la detección partiendo de lecturas cortas encontramos una insuficiente cobertura en las bases correspondientes a la unión entre exones, lo cual implica que en genes poco expresados no podamos detectar los eventos de splicing debido a esta falta de cobertura (Chaisson et al., 2015). Otro de los problemas que encontramos es que la secuencia de una misma lectura en ocasiones puede estar presente en múltiples isoformas, lo que lleva a dificultades al asignar estas lecturas a las distintas isoformas, resultando en un sesgo a la hora de detectarlas. En definitiva, la imposibilidad de obtener un transcrito de longitud completa con la tecnología de lectura corta limita el proceso de detección de splicing (Park et al., 2018). Por ello en los últimos años se está empezando a apostar por tecnologías de secuenciación de secuencia larga como Oxford Nanopore (ONT) (Bayega et al., 2021)o Pacific Biosciences (PacBio) (G. Zhang et al., 2019), los cuales consiguen en una sola lectura cubrir la totalidad de un transcrito. De esta manera se evitan muchos de los problemas antes expuestos, obteniendo una buena cobertura que permite distinguir perfectamente los diferentes transcritos,

acercándonos así más a obtener el transcriptoma completo. En la figura 2 encontramos una explicación gráfica de los problemas derivados del uso de lecturas cortas, así como la forma en que las lecturas largas lo solucionan.

Cuando hablamos de analizar un transcriptoma completo, el cual puede estar compuesto por cientos de miles de secuencias no basta solo con la secuencia de nucleótidos para poder detectar estos eventos de splicing. Se requieren herramientas bioinformáticas que permitan el procesamiento de estos datos para la detección a gran escala de estos (Muller et al., 2021).

Existen una gran cantidad de software que nos permite la detección de eventos de splicing, basados en una gran variedad de diferentes métodos (Muller et al., 2021). Muchos de estos están pensados específicamente para lecturas cortas, ya que su forma de detección pasa por alinear las lecturas de RNA obtenidas contra un genoma con exones anotados, con el objetivo de identificar los diferentes exones expresados (Min et al., 2015) .

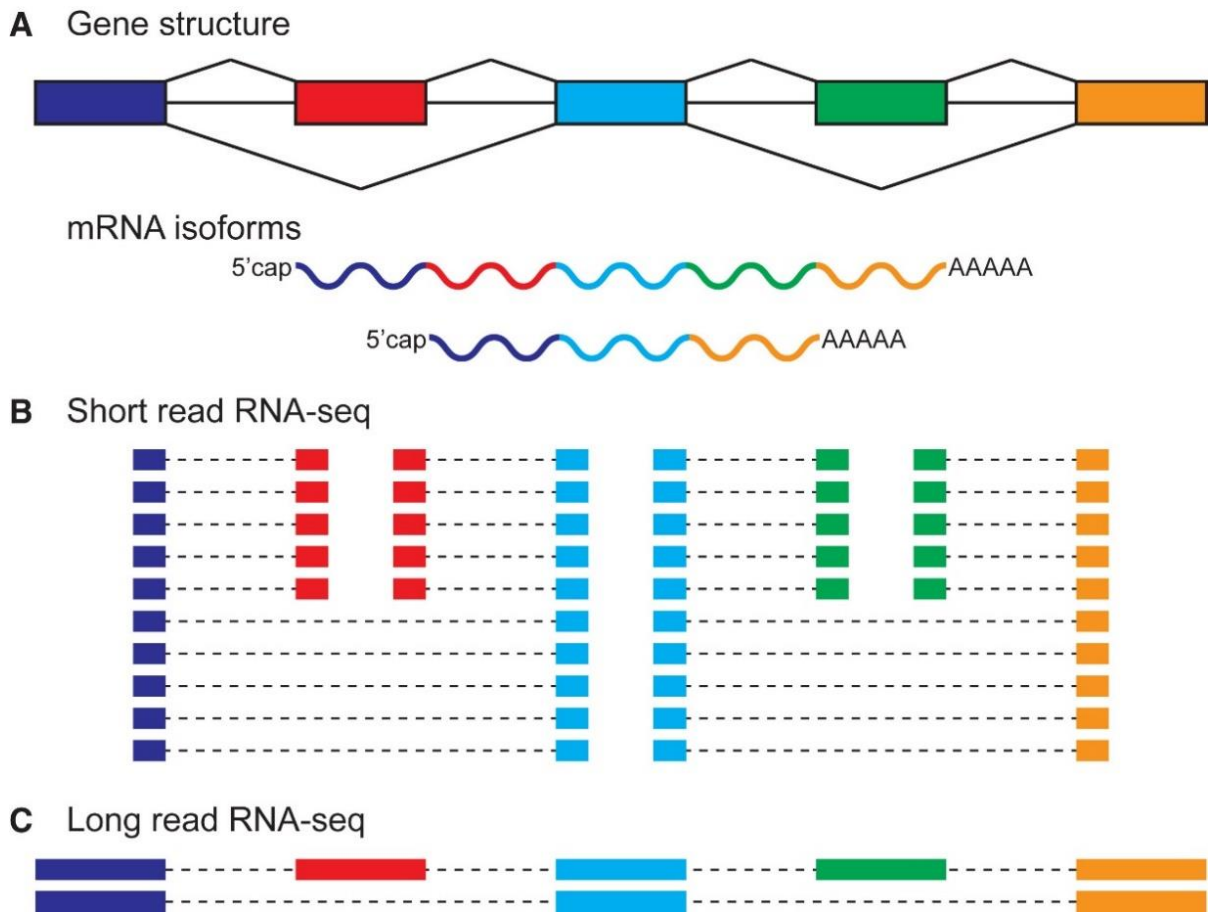

Figura 2. Ventajas y desventajas de las tecnologías de lectura corta y larga. (A) Representación de los exones que componen un gen, así como las dos isoformas que se generan a partir de este. (B) Las tecnologías de secuencia corta generan muchas lecturas, lo que facilita la cuantificación, sin embargo, no abarcan la totalidad del transcrito, por lo que se requiere procesamiento bioinformático. (C) Las lecturas largas son capaces de abarcar la totalidad del transcrito, evitando ambigüedades. Imagen tomada de (Park et al., 2018)

Un ejemplo de esto es ERANGE (Mortazavi et al., 2008) o TopHat (Trapnell et al., 2009), dos predictores muy utilizados. Sin embargo, también existen predictores cuya metodología parte de secuencias de transcritos de longitud completa, por lo que son muy aptos para ser usadas con lecturas largas. Por ejemplo, el paquete de R SpliceR (Vitting-Seerup et al., 2014) compara los transcritos con un pre-mRNA teórico obtenido a partir de los exones e intrones de cada gen, de esta forma detecta los diferentes eventos de splicing. El software SUPPA, utiliza un enfoque similar (Alamancos et al., 2015). Además, se destacan aquellas herramientas que utilizan el aprendizaje automático (en inglés machine learning) para realizar las predicciones. Este tipo de programas está en auge y ya existen varios que han mostrado buenos resultados (Louadi et al., 2019; Riepe et al., 2021)

Es importante destacar que, en general, la mayoría del software disponible para la detección de eventos de splicing necesita información del genoma para realizar las predicciones. No solo eso, si no que muchos de estos programas necesitan un genoma de buena calidad, incluso con los exones e intrones anotados para poder funcionar correctamente. Esto no supone un problema en estudios de enfermedades humanas, estudios con especies modelo como *Arabidopsis thaliana* (Kaul et al., 2000) *o C.elegans* (Consortium*, 1998), o estudios sobre especies muy bien caracterizadas que disponen de genomas de referencia bien anotados y secuenciados. Sin embargo, supone una problemática difícilmente salvable para el estudio de otras especies con menos información genómica disponible, que son la mayoría.

A pesar de que hoy día existe una gran cantidad de genomas de referencia para una diversa variedad de especies, aún existen muchas especies de interés que no disponen del mismo. Además, no todos estos genomas de referencia tienen la suficiente calidad y continuidad como para ser utilizados por los programas antes expuestos, pues algunos presentan zonas mal ensambladas, poca continuidad (están divididos en trozos denominados contigs) o falta de información en algunas zonas difíciles de secuenciar. Todo esto puede llevar a detecciones inadecuadas que imposibilitan o hacen poco recomendable utilizar estos genomas de referencia para la predicción de splicing. Por ello, surge la necesidad de desarrollar nuevas herramientas que no dependan de la comparación con el genoma para la predicción.

Existen algunos precedentes de programas que no utilizan genomas de referencia para detectar lo eventos. Un ejemplo es el método descrito en (Liu et al., 2017) que genera grupos de transcritos y comparándolos entre ellos consigue distinguir eventos de splicing sin depender de la referencia. Sin embargo, el método no es capaz de distinguir entre los diferentes tipos de splicing, solo indica la presencia de un evento.

En este trabajo de fin de grado se expone y valida un nuevo pipeline que pretende suplir esta necesidad, pues su objetivo principal es detectar eventos de splicing, así como clasificar las diferentes isoformas en función de estos partiendo tan solo de las secuencias.

1.3. Métodos de Validación

Un aspecto fundamental del desarrollo de software bioinformático; sea del tipo que sea; es el testeo, verificación y validación de éste. La alta proliferación de software bioinformático ha permitido desarrollar la genética y hacer que se adapte a las nuevas necesidades humanas, sin embargo, el rápido crecimiento en poco tiempo implica un riesgo en cuanto a los posibles errores asociados a software mal diseñado (Kamali et al., 2015). Pequeños errores en estos pueden llevar a conclusiones biológicas erróneas e incluso influir negativamente en el desarrollo posterior de diseños experimentales. Por ello, la correcta validación del software científico es crítica (Alden & Read, 2013; Wilson et al., 2012).

El proceso de validar un software implica comprobar o demostrar que cumple con las funciones para los cuales se ha diseñado. Existen varias formas de hacer esto, pero habitualmente es un proceso complicado

y que puede llegar a consumir el 50% del tiempo de desarrollo (Kamali et al., 2015). La situación más simple es cuando existe una "oráculo", es decir, algún método o mecanismo que nos permita comprobar que el *output* es correcto: Por ejemplo, en un programa que se encarga de ordenar números de mayor a menor podríamos validar el programa simplemente comprobando que el último número es más pequeño que el penúltimo y sucesivamente (Kamali et al., 2015b). Sin embargo, tal y como ocurre con frecuencia en el software bioinformático a veces es imposible disponer de un "oráculo", o este es demasiado caro para usarlo en las proporciones necesarias; esta situación se denomina "problema del oráculo" (Oracle problem en inglés). En estos casos hay que recurrir a otros métodos de validación más complejos, que nos permitan llegar a conclusiones respecto a la validez. Ejemplos son el testeo metamórfico (Giannoulatou et al., 2014), o la programación de n-versiones, en inglés N-version programming (NVP) (Yang et al., 2017).

Sin embargo, el método más común para la validación es utilizar una serie de "casos especiales" como input, es decir, utilizar sets de datos de los cuales por diferentes motivos se conoce el resultado que debería dar el software (Kamali et al., 2015), es decir se dispone de unos resultados verdaderos o *ground truth*. Por ejemplo, en la validación de un software que detecta variables estructurales se puede usar un set de datos que contenga variantes estructurales experimentalmente verificadas y comprobar que estas están siendo detectadas (Poplin et al., 2018). Es decir, comparamos el *output* obtenido con el esperado, cualquier falta de concordancia se considera un error en el software. Son incontables las ocasiones en las que este método ha sido utilizado, especialmente en aprendizaje automático, también en otros predictores de eventos de splicing (Li & Durbin, 2009; Poplin et al., 2018).

La principal limitación a este tipo de validación es la dificultad para encontrar estos casos especiales para ciertas aplicaciones, así como también la dificultad para comparar los *output*s obtenidos con los esperados (por pequeños cambios en las secuencias, diferentes nomenclaturas, errores en las anotaciones, etc.). Por ello es común utilizar datos simulados en vez de datos reales, con el objetivo de facilitar el proceso de validación (Alamancos et al., 2015).

Las diferentes nomenclaturas de clasificación y la escasez de programas dedicados a la detección de splicing alternativo sin genoma de referencia imposibilitan el uso del NVP para este caso, mientras que la complejidad de los datos dificulta establecer las relaciones necesarias para el método Metamorphic testing, así como la transformación y manejo de los inputs. Por ello, en este caso, se ha decidido utilizar la validación por caso especial como método, al ser más sencillo y estar ampliamente comprobado para estos casos.

2. OBJETIVOS.

El objetivo general de este trabajo es la evaluación de una herramienta bioinformática (en concreto un *pipeline*) para la detección de splicing alternativo cuando no existe un genoma de referencia bien anotado. (deberías decir de donde sale esta herramienta). En concreto los objetivos son.

- Optimizar la herramienta, buscando un mejor funcionamiento en cuanto a tiempo consumido y a usabilidad.

- Validar la herramienta, determinando su capacidad de distinguir correctamente entre distintos tipos de splicing

- Evaluar su utilidad en un contexto de uso real.

Para ello se proponen los siguientes objetivos específicos:

- Evaluar la estructura de las diferentes funciones del código, detectando los errores presentes en ellas.

- Modificar el código del programa, corrigiendo los errores detectados y buscando hacer más rápida la herramienta.

- Recopilar un set de datos reales para la validación del método a partir de las bases de datos disponibles, procesarlos y desarrollar una forma de comparar los eventos reales y los predichos.

- Obtener transcritos completos a partir de un proyecto de secuenciación de ARN con lecturas largas y reconstruir secuencias consenso a partir de estos.

- Obtener todos los análisis posibles a partir de los *output*s que proporciona la herramienta, buscando aprovechar al máximo el potencial de esta.

3. MATERIALES Y MÉTODOS

3.1. DESCRIPCIÓN DEL *PIPELINE*.

3.1.1. Funcionamiento general.

La funcionalidad principal del *pipeline* es predecir eventos de splicing y clasificar las diferentes isoformas a partir de lecturas de secuenciación largas sin la necesidad de recurrir a un genoma de referencia. Para ello se recurre a una estrategia basada en la reconstrucción de la parte codificante del genoma, utilizando como molde el transcriptoma. Posteriormente se mapean los transcritos contra este genoma reconstruido con el objetivo de obtener una serie de fragmentos en los que poder dividir cada locus genético. A partir del *output* de este mapeo (un archivo GTF) se detectan los diferentes eventos de splicing, estos eventos de splicing son asignados a los diferentes fragmentos obtenidos del mapeo.

Una vez se dispone de los fragmentos que componen cada locus y de los eventos de splicing asociados a los mismos se pasa a la clasificación de las diferentes isoformas, en función de los fragmentos que tengan (y los eventos presentes en los mismos). Para tener una referencia contra la que poder comparar y clasificar las isoformas se elige una isoforma representativa de cada locus, que llamaremos BLT (del inglés Baseline Transcript). Las categorías posibles para una isoforma son las siguientes: Retención intrónica (RI) si tiene un intrón retenido; Exclusión de fragmento exónico (en inglés exonic fragment exclusión, EFE) si le falta alguno de los fragmentos exónicos del BLT; Inclusión de fragmento exónico en el caso contrario (EFI); Exclusión de sitio de splicing 3’ (aceptor) si no tiene el fragmento comprendido entre dos sitios 3’ y el BLT si que lo tiene, inclusión en el caso contrario (A5E, A5I); Exclusión o inclusión de sitio de splicing 5’ (donador) si ocurre la misma situación anterior pero en el sitio 5’ (A5E/I). Estas categorías se denominan cambios estructurales y además se anota en que zona del transcrito ocurre este cambio (CDS, UTR 3’ o UTR 5').

También se analizan otros aspectos de las isoformas como cuál es el primer y el último exón (si es igual que el BLT o tiene un inicio alternativo), si el transcrito es codificante o no y su longitud. En la figura 3 encontramos el flujo de trabajo seguido por el programa. Todo el código está escrito en Python en su mayoría, excepto algunas partes en R. En la Figura 3 encontramos una representación esquemática del flujo general de trabajo del *pipeline.*

3.1.2. Reconstrucción con COGENT de los locus génicos.

Como se ha comentado anteriormente el primer paso es la reconstrucción de la parte codificante del genoma a partir de los transcritos. Para ello se utiliza la herramienta COGENT (Cogent v8.0, https://github.com/Magdoll/Cogent). COGENT es una herramienta bioinformática que toma como input un archivo FASTA que contiene transcritos de longitud completa obtenidos a partir de secuencias largas. COGENT tiene varios parámetros que se pueden cambiar en función del objetivo final y el tamaño del archivo FASTA. Puesto que nuestro objetivo es reconstruir los locus y el tamaño de los proyectos transcriptómicos usualmente van a exceder las 20.000 secuencias, utilizamos la serie de comandos descrita como “Running Family Finding for a large dataset”. En primer lugar, utilizando la herramienta de mapeo de secuencias minimap2 (tienes que poner la referencia a COGEN y a minimap2) (con el comando ‘run_preCluster.py --cpus=20’) COGENT pre-agrupan los diferentes transcritos en grupos de transcritos muy similares que potencialmente provienen de un mismo gen o familia génica. Después, utilizando MASH (pon la referencia) en cada grupo de transcritos, cada secuencia se divide en fragmentos más pequeños llamados K-mers. Comparando cuantos K-mer comparte cada par de secuencias se puede encontrar fácilmente que transcritos se parecen más entre ellos, obteniendo así los grupos de transcritos finales (cada grupo debería corresponderse con las isoformas de un mismo gen, aunque en realidad suele tratarse de familias génicas). Esto se hace con 2 comandos: run_mash.py -k 30 --cpus=12 que divide las secuencias en K-mers de 30 pares de bases (el tamaño es predeterminado, pero se podría modificar en caso de un mal resultado). Una vez obtenidos los k-mers se busca que secuencias son más similares con el comando process_kmer_to_graph.py.

Una vez divididos los transcritos en genes o familias génicas, suponiendo que en estas familias solo hay 1 gen, COGENT reconstruye la secuencia codificante del locus original tomando como base las isoformas. En esta reconstrucción, como esta basada en la secuencia del transcriptoma, solo se incluyen los exones y los intrones que estén retenidos en algún transcrito, por lo que no es una secuencia completa de locus lo que se intenta reconstruir. En la figura 4 encontramos un ejemplo esquemático de reconstrucción de un locus. Para ello se vuelve a dividir las secuencias en K-mers (los cuales son solapante), pero esta vez con el objetivo de construir un grafo de Bruijn encontrar el orden de la secuencia final.

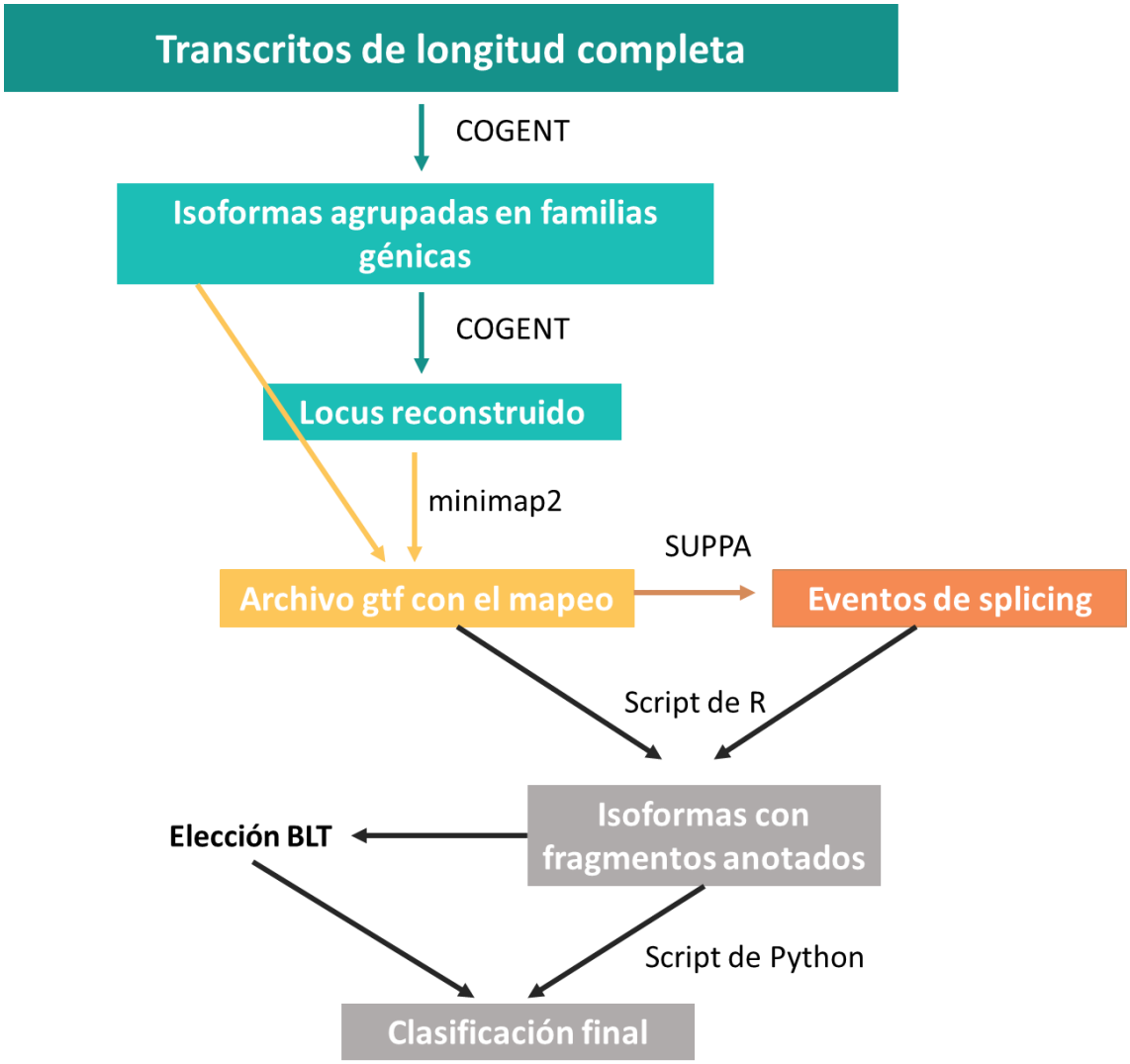

Figura 3. Esquema general del funcionamiento del pipeline. En verde se encuentran las transformaciones realizadas por el programa COGENT cuyo objetivo es agrupar los transcritos en familias génicas y reconstruir un locus. En amarillo el mapeo realizado por minimap2, cuyo objetivo es obtener una serie de fragmentos. En naranja la detección de eventos de splicing por parte de SUPPA y en gris el proceso de clasificaciones de diferentes transcritos programado en los lenguajes R y Python.

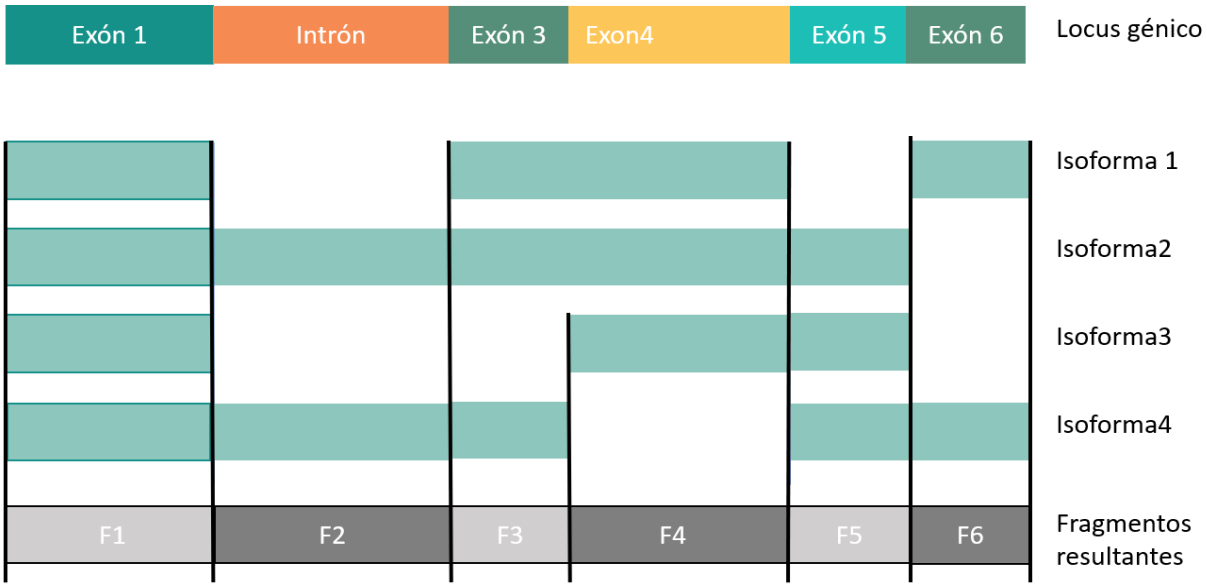

Figura 4 Detección de fragmentos mediante mapeo. Esta figura es una representación esquemática de la en la que se obtienen los fragmentos de cada locus gracias el mapeo de las isoformas contra la secuencia reconstruida. Arriba vemos la secuencia reconstruida, mientras que abajo encontramos una serie de 4 isoformas mapeadas contra este. Finalmente vemos en gris los fragmentos resultantes.

3.1.3. División del locus reconstruido en fragmentos.

Utilizando de nuevo minimap2 v2.24 (H. Li, 2018) se mapean todos los transcritos contra los locus reconstruidos, con el objetivo de obtener un archivo GTF con los diferentes fragmentos de los que se compone el locus. En la figura 4 observamos una representación esquemática del proceso de detección de los fragmentos que componen un locus mediante el uso del mapeo de secuencias. Esto se realiza con el comando "minimap2 -ax splice -t 30 --secondary=no'

Estos fragmentos no tienen por qué corresponderse exactamente con exones o intrones, puesto que en algunos casos 2 exones pueden ir en un mismo fragmento o 1 mismo exón ser separado en dos.

3.1.4. Detección de eventos de splicing

El pipeline utiliza el programa SUPPA v2 (Alamancos et al., 2015a) para detectar una serie de eventos de splicing, esto se hace a partir de la información del mapeo de minimap2. Los eventos detectables por SUPPA son retenciones intrónicas (RI), y uso alternativo de sitio aceptor/donador de splicing (A3/A5). El comando utilizado para ello es "suppa.py generateEvents". También predice otros eventos como exclusión exónica, pero estas no se usarán. Una vez SUPPA ha detectado estos eventos, la *pipeline* utiliza el paquete de R 'IRanges' v2.30.0 (Lawrence et al., 2013) para asignar cada evento de splicing detectado a los fragmentos en los que se había dividido cada locus. De esta forma los diferentes fragmentos quedan anotados según el evento de splicing que presentan. Cuando un fragmento no tiene ninguno de los 3 posibles eventos asociado se clasifica simplemente como 'fragmento exónico'.

3.1.5. Clasificación de los transcritos: categorías estructurales.

Para clasificar los transcritos el pipeline utiliza los fragmentos detectados y sus anotaciones. Para nombrar y clasificar un evento de splicing es necesario tener algo con lo que comparar, por ejemplo: un evento de exclusión exónica se define así porque un exón que está en el genoma (usamos el genoma

para comparar) no se encuentra en una isoforma. Por ello, es necesario elegir una secuencia contra la que poder comparar el resto de las isoformas, con el objetivo de poder clasificarlas en función de su splicing.

Al no disponer de la secuencia real del genoma, es mejor utilizar uno de los transcritos como referencia para el resto al cual llamaremos transcrito base (base line transcript en inglés, BLT). De esta forma si una isoforma no tiene un fragmento exónico que, sí que está en el BLT, esta isoforma será clasificada como Exclusión de fragmento exónico (en inglés exonic fragment exclusión, EFE). La única categoría que no sigue estos criterios es la retención intrónica, en ese caso siempre que haya un fragmento anotado como intrón el transcrito será clasificado como retención intrónica (RI), incluso si es el BLT. En la figura 5 se observa un ejemplo de cómo se clasifican las isoformas tomando como referencia el BLT.

Para elegir el BLT se siguen los siguientes criterios:

1. No debe tener intrones retenidos.
2. En caso de que haya más de un transcrito sin intrones, se elige el de zona codificante más grande
3. En caso de haber varios transcritos que cumplan el apartado 2, se elige aquel con las UTRs más pequeñas.
4. Si no hay ningún transcrito sin retención intrónica se elegirá como BLT aquel con mayor número de fragmentos. Tienes que explicar cómo es posible que todos los transcritos tengan intron retention.

No solo se anota la clasificación de los transcritos, sino que también se anota en que región del transcrito ocurre ese evento (CDS o UTRs).

3.1.6. Análisis de otras características de los transcritos.

Se han incluido algunos análisis adicionales para añadir información extra sobre características de los transcritos al output del pipeline. Esto proporciona un contexto más apto para los análisis, permitiendo a los potenciales usuarios analizar características como la longitud, el tamaño de la zona codificante, el tamaño de las UTRs así como si el transcrito es codificante o no.

La detección de las posiciones en las que empieza y termina la zona codificante de cada transcrito se realiza mediante la versión modificada de GeneMarkS v4.28 que permite la predicción de zonas codificantes en eucariotas y procariotas (Besemer et al., 2001). El comando utilizado es "perl GMSP_PROG -faa --strand direct --fnn --*output* o i". Este programa además nos indica si el transcrito es o no codificante, así como la longitud de su zona codificante.

Por su parte el tamaño del transcrito se determina mediante funciones básicas de Python como la función 'length'.

Figura 5. Ejemplo de clasificación de diferentes isoformas. Arriba encontramos una representación teórica de un posible BLT, las 3 secuencias siguientes son isoformas del mismo locus. Vemos como la primera isoforma tiene un intrón retenido y además le falta el fragmento correspondiente al A3 que, si tiene el BLT, por tanto, se clasifica como retención intrónica (RI) y exclusión de A3 (A3E). La siguiente isoforma tiene un fragmento A5 que el BLT no tiene, por ello se clasifica como inclusión de A5 (A5I). Por último, en la última isoforma vemos que falta el fragmento exónico3, que, si que está presente en el BLT, por ello se clasifica como exclusión de fragmento exónico (EFE).

3.1.7. *Output* general del *pipeline*:

El pipeline da como *output*s dos archivos: un archivo en los que se indica los fragmentos que tiene cada transcrito, así como las características de estos fragmentos: posiciones en el locus reconstruido, locus al que pertenecen y su anotación (fragmento exónico, A5, A3 o RI). El otro archivo recoge las características de cada transcrito, indicando su familia génica, nombre, fragmentos que lo componen, longitud, longitud de la CDS y las UTR, categorías estructurales que presenta, tipo de inicio de transcripción (como el BLT o inicio alternativo) y final de transcripción, y eventos que presenta el transcrito en cada una de sus regiones. Además, se obtienen *output*s intermedios como la anotación de SUPPA o archivos que resumen de la actuación de COGENT.

3.2. OPTIMIZACIÓN DEL PROGRAMA.

Para la optimización del *pipeline* se revisaron todas las funciones del *pipeline* en busca de errores. Para ello, se utilizaron una serie de transcritos ficticios que no contenían información de secuencia si no solo de fragmentos y tamaños (ejemplo en el material suplementario), pudiendo así controlar cual sería la clasificación correcta de cada transcrito.

Para la optimización de las funciones se buscó reducir el número de 'bucles for' presentes en cada función, así como eliminar el uso de librerías no predeterminadas de Python, como por ejemplo Python-intervals. Estas librerías fueron sustituidas por funciones básicas integradas en el lenguaje.

Para determinar la mejora en la velocidad de las diferentes funciones se calculó el tiempo que tardaba la versión original y la optimizada en procesar un set de 2000 transcritos simulados. El tiempo fue estimado con la herramienta 'time profiler' integrada en el programa Spyder v5 (Raybaut, P, 2009).

Por último, una vez completado el código y optimizadas todas las funciones se realizan varias pruebas generales con pequeños sets de datos para ver el funcionamiento general del programa. Se analizaron los diferentes *output*s, no solo la clasificación final, si no también *output*s intermedios como los resultados de SUPPA, el GTF resultado del mapeo de minimap2, buscando algún error en el funcionamiento general.

3.3. VALIDACIÓN DEL PROGRAMA

Para validar el *pipeline* se realizó una validación por caso especial, es decir, partiendo de un resultado ya conocido previamente se comprobó que el programa llegaba a este mismo resultado. A partir de esta metodología se cuantificaron los falsos positivos y negativos, así como los verdaderos positivos y negativos y finalmente se calcularon una serie de medidas que nos ayudan a cuantificar la validez de los resultados. Estas medidas se suelen definir para datos binarios (por ejemplo, presenta o no presenta retención intrónica), sin embargo, el *output* que pretendemos analizar es de múltiples clases, por lo que se debe calcular por separado para cada categoría y finalmente hacer un promedio. Las métricas elegidas son:

Ratio de falsos negativos (FNR): Definido como la probabilidad de que un positivo no sea detectado (Por ejemplo, la probabilidad de no detectar una retención intrónica como tal). Vemos la fórmula en la ecuación 1.

Ecuación 1. $$FNR = \frac{Falsos\ negativos}{Falsos\ negativos + verdaderos\ positivos}$$

Ratio de falsos positivos (FPR): Definida como la probabilidad de que un resultado positivo en realidad sea negativo (Por ejemplo, que un evento detectado como retención intrónica en realidad no lo sea). Vemos la fórmula en la ecuación 2.

Ecuación 2. $$FPR = \frac{Falsos\ positivos}{Falsos\ positivos + verdaderos\ negativos}$$

También se calcularon la sensibilidad y la especificidad, métricas complementarias a las dos primeras:

La sensibilidad se define como la probabilidad de que un positivo real sea detectado como positivo por el programa (por ejemplo, que una retención intrónica real sea detectada como tal). Vemos la fórmula en la ecuación 3.

Ecuación 3. $$Sensibilidad = 1 - FNR$$

La especificidad se define como la probabilidad de que un negativo real sea detectado como negativo por el programa (por ejemplo, que un evento que no es una retención intrónica sea detectado como cualquier otra categoría).

Ecuación 4. $$\text{Especificidad} = 1 - FPR$$

Existe una dificultad para validar la clasificación final de los transcritos, la nomenclatura utilizada (que incluye exclusiones e inclusiones dependientes del BLT) hace muy difícil predecir el resultado esperable partiendo de datos reales. Por ello, se validó el paso previo a la clasificación final, es decir, la anotación

de los fragmentos. La anotación que SUPPA da a estos fragmentos es la información básica utilizada para clasificar los transcritos, por lo que si esto es correcto podemos dar como buena la metodología.

Se realizó esta validación con dos sets de datos reales: Un set de transcritos provenientes de 1820 genes del cromosoma 2L de *Drosophila melanogaster* (mosca de la fruta) y otro de 744 genes del cromosoma 1 de *Danio zebrato* (pez cebra). Para obtener la anotación "real" o de referencia (el caso especial), se descargó el GTF conteniendo la anotación real de estos transcritos sobre sus genomas de referencia (datos del NCBI). Sobre este archivo GTF se utilizó la herramienta SUPPA, obteniendo una serie de eventos de splicing. Esto constituye el 'caso especial' que usaremos para la validación.

A partir de estos GTF y el genoma de referencia se obtuvo un archivo fasta con la secuencia de los transcritos utilizando la herramienta gffread de Cufflinks v2.2.1 (Trapnell et al., 2010) con el comando 'gffread t.gtf -referencia.fasta -w transcritos.fasta'. A partir de estas secuencias, se aplicó el *pipeline* desde el principio, obteniendo también una serie de eventos de splicing. Comparando los eventos detectados a partir del GTF y aquellos detectados a partir del pipeline se evaluó la fiabilidad del método. Lo que estamos comparando de esta manera es si, para el mismo dato, la predicción de los sitios de splicing que se puede obtener usando un genoma de referencia, que viene dado por el gtf, es la misma que si se hace sin tener esta información, sólo con el dato de la secuencia.

La comparación de los eventos obtenidos por el pipeline y los eventos de la referencia se realizó mediante un script de R que utiliza el paquete IRanges. Los fragmentos (los eventos) están identificados por su posición en el genoma, por lo que se comparan fragmentos con las mismas posiciones. Puesto que el genoma utilizado en la referencia es el real (contiene intrones, zonas no codificantes, etc) las posiciones de los fragmentos no coinciden con los fragmentos del *output* del *pipeline*. Esto se debe a que en el *pipeline* se utiliza el genoma reconstruido, que carece de zonas que no se expresan. Por ello, antes de la comparación se modifican las coordenadas de los fragmentos de referencia, eliminando las posiciones correspondientes a fragmentos que no se encuentran en los transcritos.

3.4. APLICACIÓN A UN CASO REAL: MICROPTERUS SALMOIDES

Para probar la versatilidad y funcionalidad del *pipeline* se analizó un proyecto transcriptómico real de una especie con escasa información genómica: *Micropterus salmoides* (lubina negra).

3.4.1. Datos utilizados

Se utilizó un proyecto de secuenciación con la tecnología SMRT de PacBio Systems del transcriptoma completo de *Micropterus salmoides*. Este proyecto fue cedido por un laboratorio externo. El ARN secuenciado fue obtenido a partir de una mezcla de tejidos de la gónada, la cabeza y el tronco del riñón, el estómago, el cerebro y el hígado de un pez control (sano). Por tanto, se trata de una muestra representativa de los transcritos de la especie.

3.4.2. Procesamiento de las lecturas y generación de las isoformas de longitud completa

Los transcritos de longitud completa fueron generados a partir de las lecturas iniciales utilizando el software Isoseq v3 de Pacific Biosciences. En primer lugar, se generarón las lecturas circulares consenso con el comando 'ccs fish.subreads.bam fish.ccs.bam --min-rq 0.9'. Posteriormente se eliminaron las etiquetas y los primers que contenían las lecturas, dejando así solo la secuencia de ARN. Para ello se usó la herramienta lima y el comando 'lima fish.ccs.bam primers.fasta fish.fl.bam --isoseq --peek-guess'. Posteriormente se refinaron las secuencias eliminando las colas poli-A y los concatémeros generados por la secuenciación SMRT, el comando utilizado fue 'isoseq refine fish.NEB_5p--NEB_Clontech_3p.fl.bam fish.flnc.bam --require-polya'. Finalmente, las lecturas fueron agrupadas por similitud para generar transcritos consenso que fueron anotados en un archivo fasta. El comando utilizado fue 'isoseq cluster fish.flnc.bam clustered.bam --verbose --use-qvs' del cual se obtuvieron varios archivos, entre ellos el archivo FASTA con todos los transcritos de longitud completa.

3.4.3. Clasificación de las isoformas.

La clasificación de las isoformas fue realizada por el *pipeline*, siguiendo los pasos explicados anteriormente.

3.4.4. Análisis de términos GO.

Se anotaron los términos GO de las familias génicas detectadas por COGENT. Para ello se utilizó el software Blast2GO (v6.0) (Conesa et al., 2005), el cual asigna términos GO a una secuencia en función de su similitud con otras secuencias ya anotadas.

Se analizó si había algún término GO enriquecido en los genes que presentaban cada una de las diferentes categorías de transcriptos, con el objetivo de comprobar si estos eventos de splicing estaban asociados a algún evento en particular; Esto fue realizado mediante una prueba de Fisher realizada en R. Se consideraron enriquecidos aquellos términos GO que tras la corrección de prueba múltiple mediante el método FDR (tasa de falso descubrimiento) presentaban un p-value menor a 0.05. Para el análisis se utilizaron todos los niveles de términos GO.

4. RESULTADOS

4.1. OPTIMIZACIÓN DE LAS FUNCIONES DEL *PIPELINE*.

Se analizaron tres funciones del pipeline: La primera es la que selecciona el transcrito que será usado como BLT en cada gen. Esta función elige como BLT aquel transcrito que no tenga retenciones intrónicas; en caso de que haya más de uno elige el de la ORF más larga. En caso de que varios tengan la misma ORF se elige el de UTR más corta. Sin embargo, debido a un error la función elegía como BLT cualquier transcrito que tuviese esa longitud de UTR, tuviese o no retención intrónica. Esto fue resuelto, pero no se consiguió aumentar la velocidad de la función, pues ya estaba muy optimizada.

La segunda función en ser analizada fue la que determina si una isoforma tiene o no retención intrónica. Esta función fue reprogramada para hacerla 78 veces más rápida (de 7,8 segundos a 100,3 milisegundos) tras reducir los bucles 'for' a uno solo y eliminar las funciones de la librería 'Python-intervals'. Además, se corrigió un error relacionado con la determinación de en qué zona del transcrito está la retención.

También se intervino la función que detectaba inclusiones y exclusiones exónicas. El error que presentaba es que solo detectaba si un fragmento del transcrito no se encontraba en el BLT (inclusión exónica) o si un fragmento presente en el BLT faltaba en el transcrito. Sin embargo, no tenía en cuenta la naturaleza del fragmento, es decir, cualquier retención intrónica era clasificada como inclusión de fragmento exónico también. La velocidad aumento 82 veces (función original 4 segundos, función optimizada 44 milisegundos), esto se consiguió de nuevo reduciendo el número de bucles 'for' a 1 solo (en el mismo se detectan las inclusiones y exclusiones) y eliminando las funciones de 'Python-intervals'.

Por otra parte, se añadieron al código 2 funciones más que no se encontraban en el *pipeline* original, las funciones para clasificar los cambios en los sitios donadores o aceptores de splicing, A5 y A3. Estos eventos de splicing se obtenían como parte del *output* generado por SUPPA, sin embargo, no eran utilizados para la clasificación de los transcritos por lo cual había una pérdida de información. Para incluir estas funciones se utilizó como base la función de los cambios en exones y se modificó ligeramente para que el cambio en los fragmentos del transcrito se tuviese en cuenta solo si el fragmento estaba anotado con A5 o A3.

Además, observando el GTF obtenido de minimap2 se evidenció que un 45% de los transcritos estaban mapeando en los locus equivocados. Esto generaba problemas en el funcionamiento del código generando errores, motivo por el cual se decidió mapear cada cluster de transcritos solo contra su locus en lugar de contra todo el 'genoma' (unión de todos los locus reconstruidos). Esto hizo que el 100% de los transcritos mapearan correctamente, solucionándose así los problemas derivados.

4.2. VALIDACIÓN DE LOS FRAGMENTOS ANOTADOS.

Utilizando un set de transcritos provenientes de genes reales de 2 especies modelo: Drosophila y pez cebra, se realizó un análisis de la actuación del *pipeline*. Primero, se detectaron los eventos de splicing presentes en los transcritos utilizando SUPPA y los GTF descargados del NCBI. Una vez obtenida esta información, se utilizó el *pipeline* completo partiendo de los transcritos reconstruidos, para obtener el *output* de clasificación de los fragmentos del *pipeline* a partir del locus reconstruido de COGENT. Después se compararon los eventos detectados. De esta forma, podemos crear una matriz de confusión, detectando así los falsos positivos y negativos que tiene el *pipeline* a la hora de detectar distintos eventos de splicing. En la figura 6, se observan las matrices de confusiones generadas para ambas especies. En esta imagen, los colores indican el porcentaje que representa cada tipo de evento predicho en el total de un tipo de evento real, es decir, tomando de ejemplo todos los A3 reales, cuantos de ellos han sido predichos como RI, cuantos como fragmentos exónicos, etc. Los 'fragmentos exónicos' se

corresponden con fragmentos del transcrito en los que SUPPA no ha detectado retenciones intrónicas ni cambios en los sitios aceptores/donadores de splicing.

A simple vista se puede observar como muchos eventos se están clasificando como 'fragmentos exónicos' sin serlo, además, en drosphila vemos como muchos A3/A5 se confunden con intrones retenidos; sin embargo, esto no ocurre en pez cebra. Los fragmentos exónicos son los que mayor número de verdaderos negativos tienen, aunque en el caso de Drosophila las retenciones intrónicas también han sido detectadas correctamente.

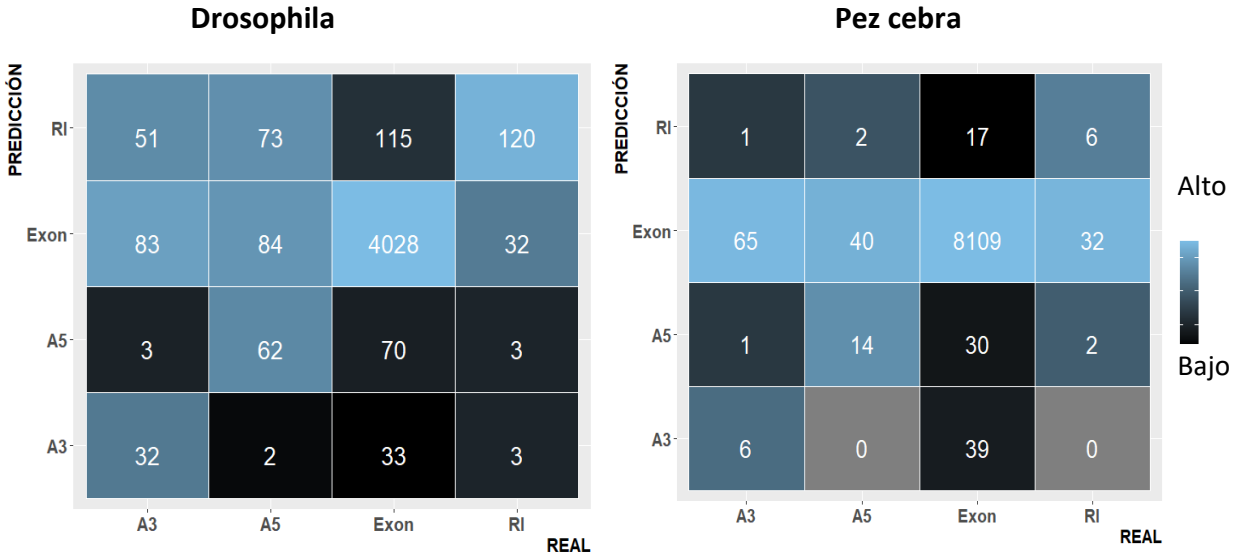

Figura 6. Matrices de confusión para los eventos de splicing. Matriz en la que se puede observar la correspondencia entre los eventos reales y aquellos predichos por SUPPA para las especies *Drosophila melanogaster* y *Danio zebrio* (pez cebra). Los colores indican el porcentaje que representa cada valor y han sido normalizados por columna, es decir, por evento real. Además la escala de color es logarítmica para evitar outliers.

Para determinar la validez de las predicciones hay que utilizar unas medidas más exactas que los datos en bruto, por ello a partir de estas matrices de confusión se han calculado los falsos positivos y negativos, así como los verdaderos positivos y negativos. Estos valores a su vez se han utilizado para calcular una serie de métricas que nos permitan saber con exactitud la validez del *pipeline*. Estas son sensibilidad, medida que indica la probabilidad de que un evento sea predicho y clasificado correctamente; especificidad, que nos indica la probabilidad de que un evento que no es de un tipo concreto no sea clasificado como tal (si un evento no es una retención intrónica, la probabilidad de que el *pipeline* lo clasifique como cualquier cosa que no sea retención intrónica); el ratio de falsos negativos (FNR): Definido como la probabilidad de que un positivo no sea detectado (Por ejemplo, la probabilidad de no detectar una retención intrónica como tal); y el ratio de falsos positivos (FPR).: Definida como la probabilidad de que un resultado positivo en realidad sea negativo (Por ejemplo, que un evento detectado como retención intrónica en realidad no lo sea).

En la figura 7 vemos los resultados obtenidos en estas métricas para ambas especies, desglosadas en las diferentes categorías posibles. Vemos como existe una ratio de falsos positivos muy baja en todos los casos (menor del 5%), excepto en los fragmentos exónicos (siendo del 36 y 81% en Drosophila y Pez cebra respectivamente). Esto se ve reflejado en una alta especificidad (mayor al 90%) en todos los casos menos en este último.

En general, podríamos decir que el programa presenta una ratio de falsos positivos bajo y una ratio de falsos negativo moderadamente elevada. Esto se refleja en valores muy altos de especificidad (en valores promedio 89 y 79% respectivamente para Drosophila y pez cebra), pero unos valores no muy altos de sensibilidad (en promedio 46 y 36% respectivamente).

También se han analizado estas métricas cuanto a la capacidad de detectar eventos de splicing sin tener en cuenta su categoría, (en otras palabras, si los eventos de splicing se detectan como tal ya sea A3/A5 o RI, o se están anotando como si fuese un fragmento exónico). La sensibilidad del programa para detectar un evento es del 58% en pez cebra y del 78% para Drosophila, mientras que la especificidad es 77 y 84% respectivamente. Por su parte la precisión es del 92 y 96% respectivamente. Estos valores indican que el *pipeline* es capaz de distinguir correctamente la presencia o ausencia de eventos de splicing.

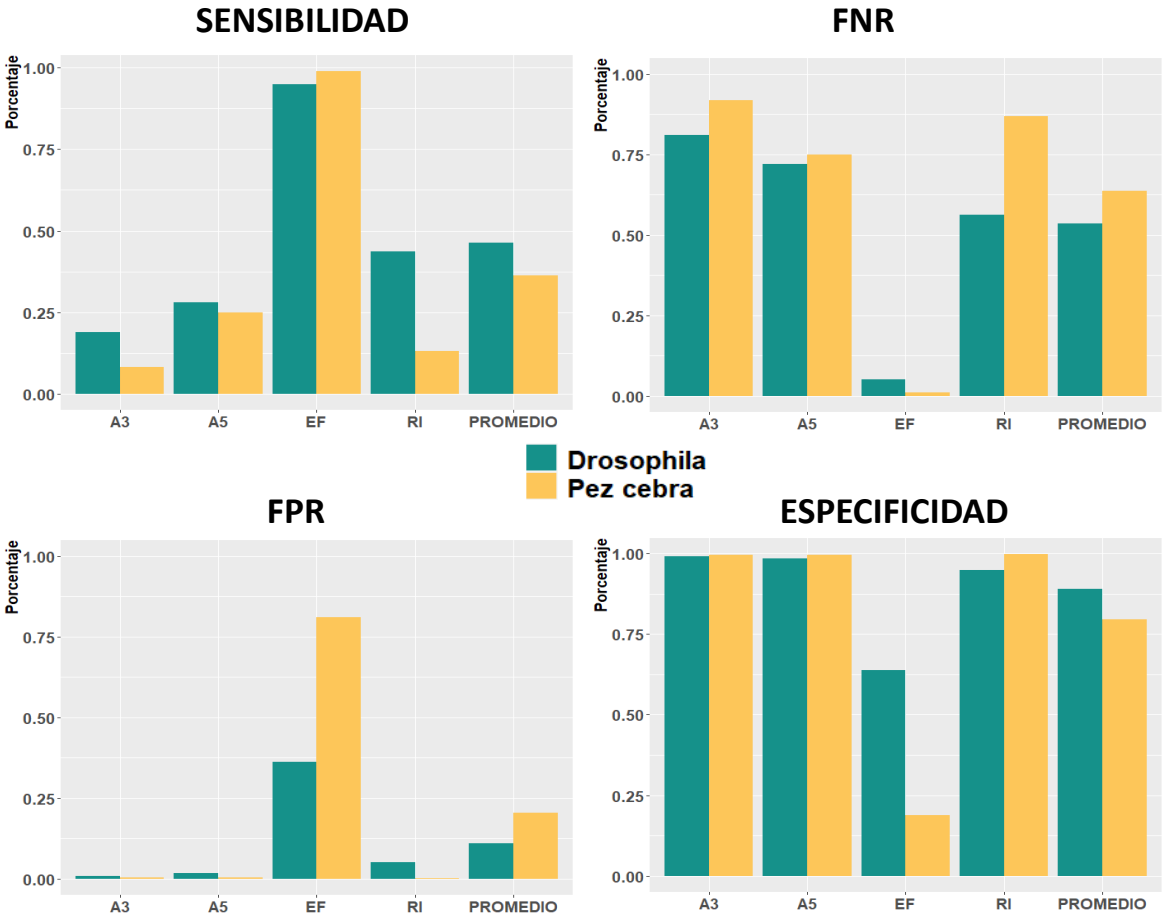

Figura 7 Resultados de la validación en datos reales. En estas gráficas vemos los valores de FNR, FPR, sensibilidad y especificidad obtenidos por el pipeline en la anotación de los fragmentos. Se han calculado estos valores para cada una de las posibles anotaciones, incluyendo los fragmentos exónicos (EF). Estos valores han sido calculados a su vez para dos especies diferentes: Drosophila y Pez cebra.

4.3. APLICACIÓN A UN CASO REAL: *Micropterus salmoides*

4.3.1. Reconstrucción de la secuencia codificante

Como primer paso del "*pipeline*" se utiliza la herramienta COGENT, esta agrupa todos los transcritos en grupos de transcritos (en inglés "clusters") que representan posibles familias génicas. Puesto que cada familia génica contiene varios transcritos, COGENT intenta reconstruir una secuencia consenso que teóricamente contenga toda la secuencia codificante.

En este caso, tras el procesamiento del dataset de *Micropterus Salmoides* se ha obtenido un archivo FASTA con 36852 transcritos únicos. COGENT ha conseguido agrupar 22857 de estos transcritos en 7322 familias génicas. Esto implica que hay un 36% de las secuencias que no se han incluido en ninguna familia génica. El tamaño de estas familias génicas es variable, sin embargo, la mayoría de ellas contienen de 1 a 5 transcritos, aunque dentro de este grupo destacan aquellas con de 1 y 2 isoformas (Figura 8A).

Además, COGENT ha conseguido reconstruir un único contig en buena parte de estas familias génicas (Figura 8B), teniendo la mayoría de ellas menos de 3 reconstrucciones. Sin embargo, también encontramos familias génicas con un número alto de contigs que podrían dar lugar a problemas en la predicción de eventos de splicing.

Como se aprecia en la figura 8C el número de reconstrucciones tiende a aumentar con el número de isoformas en la familia génica, con un coeficiente de correlación de 0.87. A más isoformas aumenta la posibilidad de que existan ambigüedades dando lugar a una mala reconstrucción de la secuencia codificante, lo cual se traduce en más contigs. En ocasiones puede haber situaciones muy complejas en las que las ambigüedades sean tantas que no se consiga reconstruir ninguna secuencia; una manera de resolver este problema y evitar las ambigüedades puede ser aumentar el tamaño de K-mer, sin embargo, en esta ocasión no ha sido necesario.

4.3.2. Detección de fragmentos exónicos y eventos de splicing.

Tras la reconstrucción se mapean usando minimap2 los transcritos contra estas secuencias para detectar los diferentes potenciales fragmentos exónicos para cada gen. En la figura 9A se observa la distribución del número exones frente al número de genes, teniendo la mayor parte de estos entre 1 y 5 exones. También encontramos 278 genes con un solo exón, un 3,8% del total de genes. En total se han detectado 39493 exones, por lo que la media de exones por gen es de 5,394.

Como ya se ha comentado, SUPPA predice los eventos de splicing presentes en los transcritos a partir del mapeo de estos contra el locus, esta información es utilizada para anotar los fragmentos que componen los transcritos. En la figura 9B podemos ver como la gran mayoría de los fragmentos (96%) se corresponden con fragmentos exónicos sin ningún tipo de evento en ellos mientras que en el 4% restante, la mayoría de los eventos se corresponden con retenciones intrónicas, representando el uso alternativo de sitios aceptores/donadores de splicing (A3 y A5) un menor porcentaje del total.

Como ya se ha comentado, SUPPA predice los eventos de splicing presentes en los transcritos a partir del mapeo de estos contra el locus, esta información es utilizada para anotar los fragmentos que componen los transcritos. En la figura 2B podemos ver como la gran mayoría de los fragmentos (96%) se corresponden con fragmentos exónicos sin ningún tipo de evento en ellos mientras que en el 4% restante, la mayoría de eventos se corresponden con retenciones intrónicas, representando el uso alternativo de sitios aceptores/donadores de splicing (A3 y A5) un menor porcentaje del total.

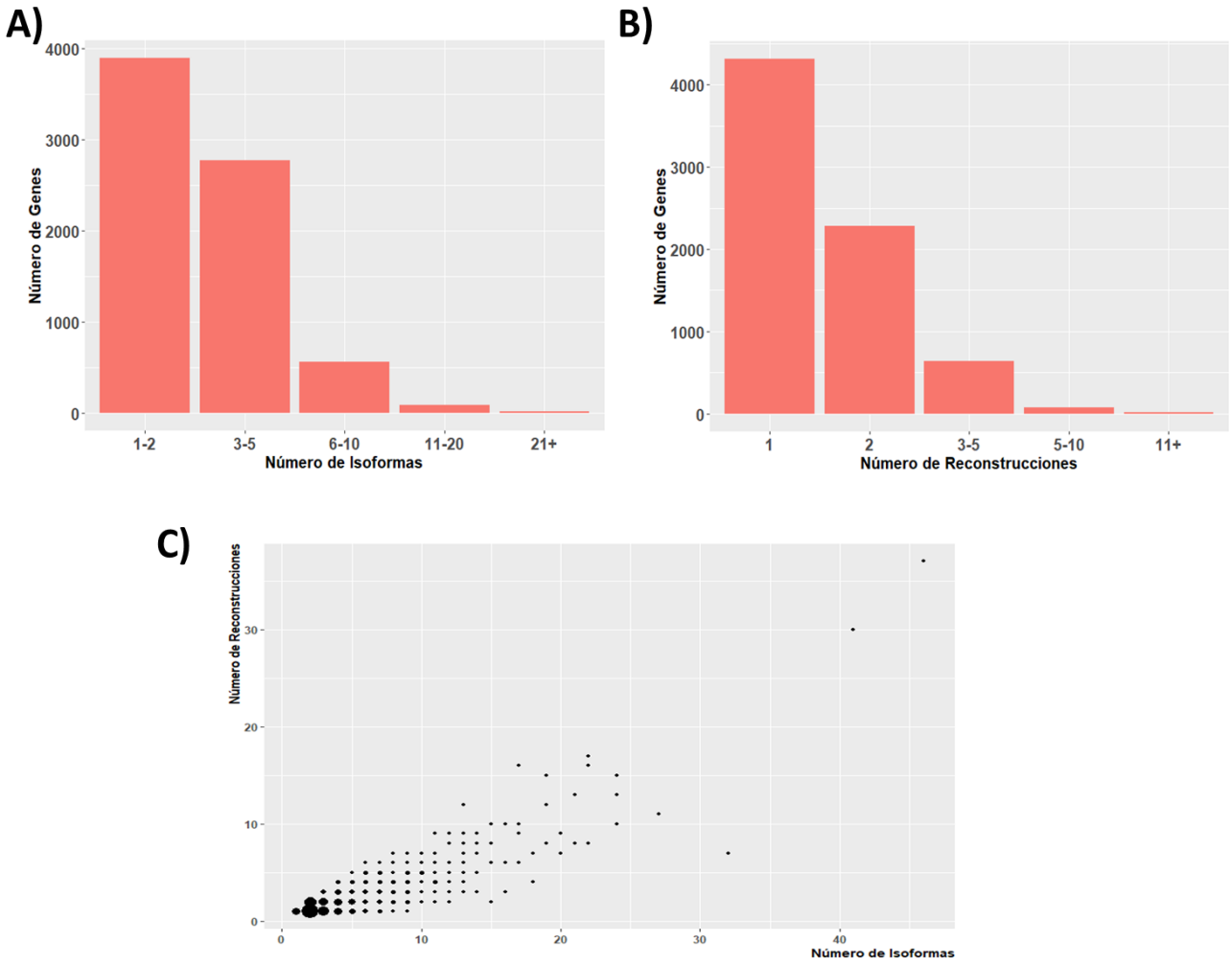

Figura 8. Resultados de COGENT. A) Distribución del número de isoformas por gen, la mayor parte de los genes cuentan con de 3 a 5 transcritos. **B)** Distribución del número de reconstrucciones por gen, el grupo más numeroso es el de los genes que han resuelto la reconstrucción en una sola reconstrucción, sin embargo, también se observan un número considerable de genes con 2 o más reconstrucciones. **C)** Correlación entre el número de Isoformas que presenta un gen y su número de reconstrucciones. Estas dos variables muestran una correlación positiva con un coeficiente del 0.87.

A)

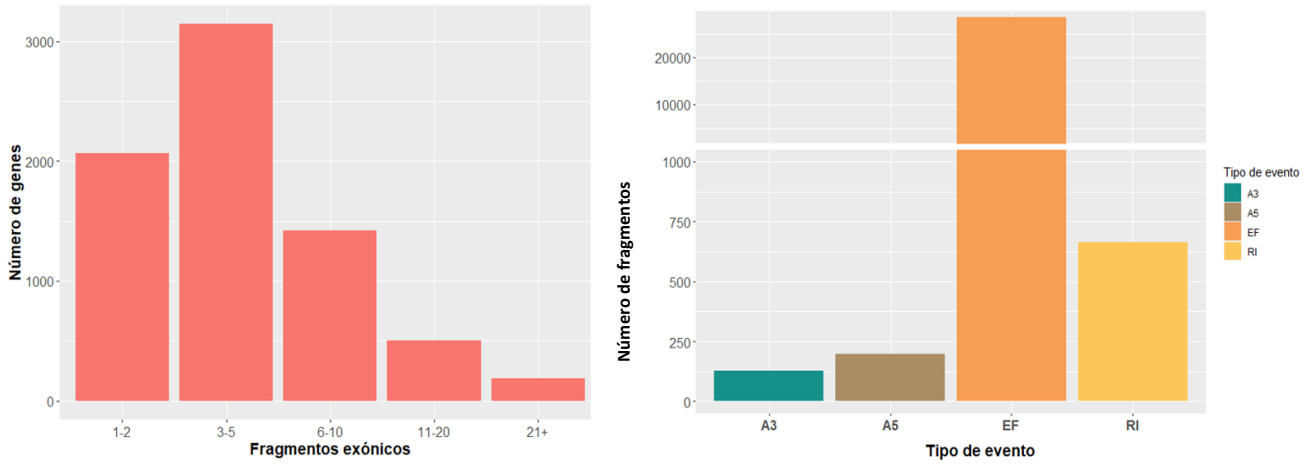

Figura 9. Distribución del número de exones y los eventos de splicing. A) Distribución del número de fragmentos exónicos (EF) por gen.. **B)** Distribución de los eventos de splicing en los fragmentos exónicos, la mayor parte de los exones no tienen RI, A5 o A3 asociadas, mientras que entre estas categorías la más común es la retención intrónica.

4.3.3. Características de los transcritos no relacionadas con el splicing.

Existen varios tipos de transcritos y no todos tienen como función la expresión proteica; muchos de ellos tienen funciones regulatorias, de inhibición o defensiva. En este caso se utilizó la herramienta GeneMarkST para predecir el potencial codificante de cada transcrito. De los más de 22.857 transcritos, la mayoría (87%) de los transcritos fueron predichos como transcritos codificantes, y tan solo 2853 transcritos no codifican para ninguna proteína, representando un 12% del total (Figura 10A). Estos transcritos tienen un rango de longitudes desde los 114 hasta los 6000 pb, con una longitud media de 2287pb, concentrándose en su mayoría entorno a los 300-1000 pb y entorno a los 3000-4500 pb. Por comparación, vemos como los transcritos codificantes presentan un rango desde los 323 a los 11750 pb, siendo su longitud media de 3186 pb y concentrándose sobre todo en torno a los 3000-4500 pb (Figura 10B y 10C).

Como vemos los transcritos no codificantes son en general más pequeños que aquellos que codifican para alguna proteína, tanto en rango de tamaños como la media. Sin embargo, los transcritos no codificantes presentan una mayor variabilidad en el tamaño debido a que se concentran en torno a dos picos de tamaño, siendo este segundo pico solapante con el correspondiente a los transcritos codificantes.

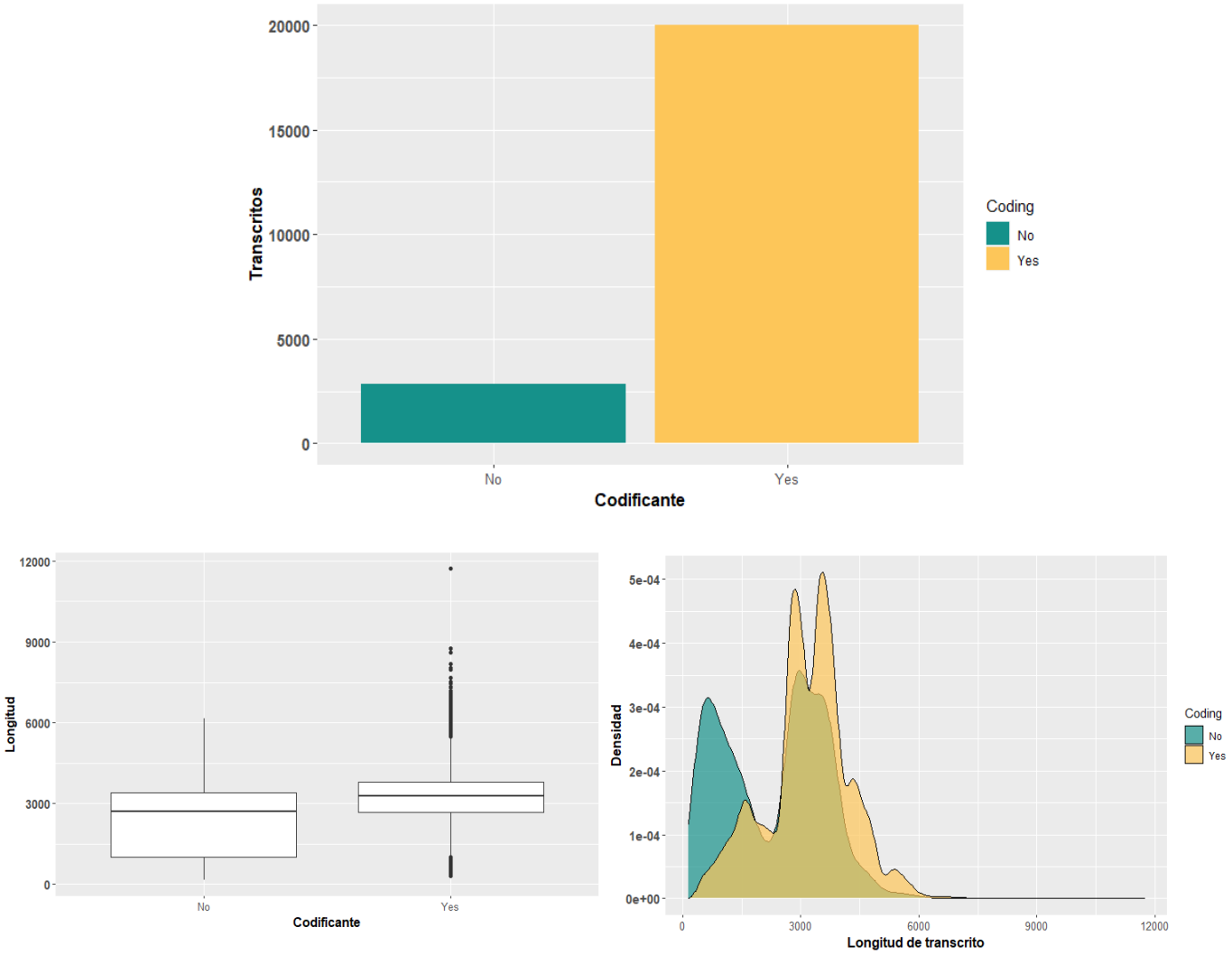

Figura 10. Características genómicas de los transcritos noc. A) Distribución de los transcritos codificantes y no codificantes, la mayor parte de los transcritos son codificantes. **B)** Distribución de longitudes de ambos tipos de transcritos mediante un gráfico de densidad, los transcritos codificantes se concentran entorno a los 3000 y 4500 pb, mientras que los no codificantes aparecen concentrados en torno a 2 longitudes, entorno a 300-1000 pb y entorno a los 3000 y 4000 pb.

4.3.4. Clasificación de los transcritos en función de sus eventos de splicing.

Una vez se han detectado los eventos de splicing y se han inducido las principales características de los transcritos se clasifican estos en función de los eventos de splicing que presentan. A la hora de clasificar cada transcrito de un gen concreto utilizamos uno de ellos como referencia contra la que comparar al resto. Este se denomina Baseline transcript (BLT), elegido según las características explicadas anteriormente.

En la figura 11A observamos la distribución de los transcritos en sus diferentes categorías. Hay 7322 transcritos clasificados como BLT, lo cual coincide con el número de genes detectados (hay un BLT por gen), mientras que las categorías más representadas son aquellas que se corresponden con cambios exónicos (EFI y EFE con 9110 y 7639 transcritos respectivamente), también destaca la retención intrónica (1435 transcritos) aunque en menor medida. La inclusión y exclusión de sitios alternativos

están poco representadas (todos por debajo de los 400 transcritos), al igual que los transcritos que son la única isoforma en su gen (365 transcritos).

Si nos fijamos en el nivel génico (Figura 11B), es decir no distinguimos entre inclusión o exclusión, vemos como la mayoría de los genes tienen transcritos con cambios en los fragmentos exónicos (El 78%). Estos están seguidos en menor cantidad por aquellos genes que presentan una retención intrónica en algún transcrito. Los menos numerosos son aquellos en los que varían los sitios de splicing (A3/A5).

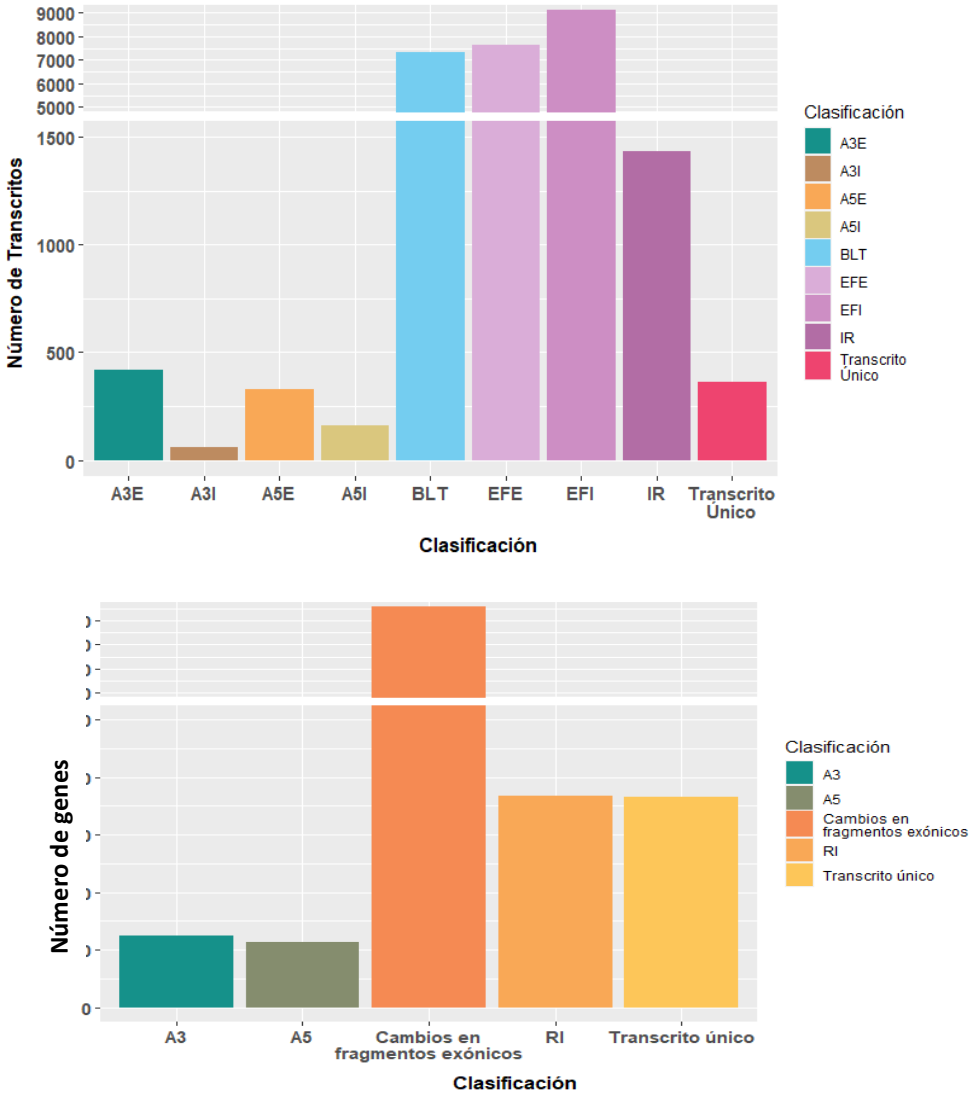

Figura 11. Distribución transcriptómica y génica de las diferentes categorías de clasificación. A) Número de transcritos que ha sido clasificado como cada una de las categorías disponibles en el pipeline: BLT (transcrito base), Transcrito único (el único transcrito del gen), EFE/EFI (exclusión o inclusión de fragmentos exónicos respectivamente), A3E/I(exclusión o inclusión de sitio aceptor alternativo), A5E/I (exclusión o inclusión de un sitio donador alternativo) e IR (transcritos con una retención intrónica). B) Número de genes que presentaban alguna de las categorías de splicing (sin tener en cuenta inclusiones ni exclusiones): A3,A5, cambios en los fragmentos exónicos, RI y transcrito único (para los genes que solo tienen un transcrito).

También es interesante fijarse en las regiones del transcrito en las que ocurren estos cambios más frecuentemente. En la figura 12 podemos ver la distribución transcriptómica de los distintos eventos de splicing en función de la región del transcrito en la que ocurre. La presencia de cambios exónicos es más común en las regiones UTR que en la CDS, mientras que las retenciones intrónicas fueron más comunes en la CDS que en las UTR. Los cambios en los sitios aceptores/donadores de splicing parecen estar al mismo nivel en las distintas regiones.

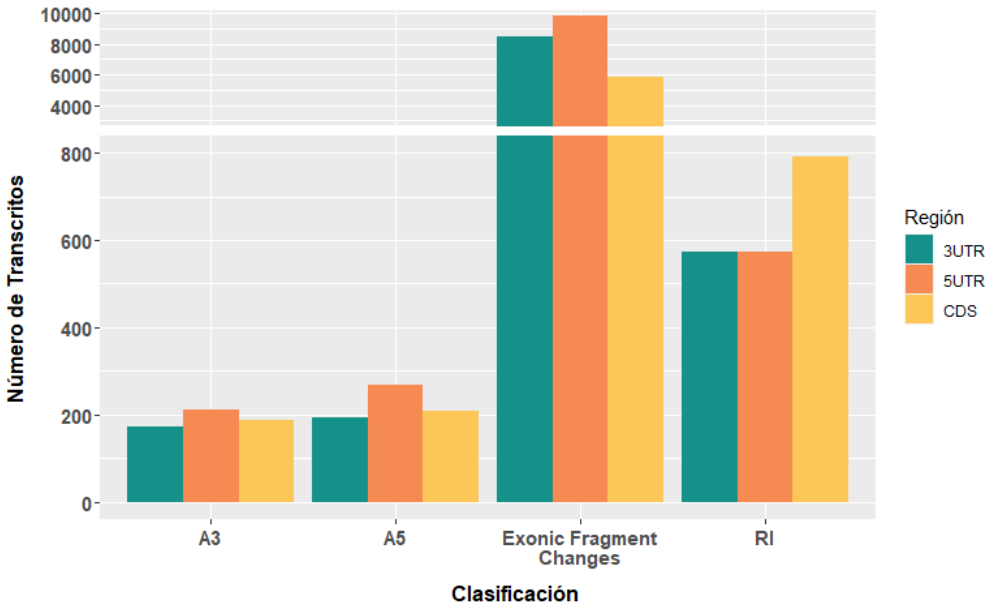

Figura 12. Distribución de los eventos de splicing en las diferentes regiones transcriptómicas. Distribución transcriptómica de las diferentes categorías en función de la región del transcrito en la que se encuentran: en la zona codificante (CDS) o en las UTR.

Dentro de los cambios en fragmentos exónicos hay un tipo en especial; aquellos cambios que afectan al sitio de inicio o final de transcrito (en ingles Transcription terminal site TTS y Transcription start site TSS respectivamente). Estos son especialmente relevantes pues afectan a la regulación transcripcional. Cerca del 60% de los transcritos tienen un sitio de inicio diferente al del BLT; sin embargo en el caso de los sitios de finales de transcrito esta tendencia está invertida, siendo más común que los transcritos terminen como el BLT, lo cual es lógico por cómo funciona la tecnología, que selecciona los transcritos por la cola poliA.

Puesto que el BLT no es más que un transcrito elegido para comparar al resto de transcritos de un mismo gen es más informativo analizar los sitios de inicio y de final a nivel génico y no de transcrito. En la figura 13 se observa cómo más del 60% de los genes tienen todos los transcritos con el mismo sitio terminal, mientras que algo menos del 30% tiene algún transcrito con final alternativo. Esto mismo ocurre para el sitio de inicio, pero la diferencia es menos acusada. Además, los genes cuyos transcritos tienen sitios de inicio de transcrito alternativos (TSS) tienden a tener también sitios de final de transcrito (TTS) alternativo (Test de Fisher, pvalue=2.2×10^{-16}).

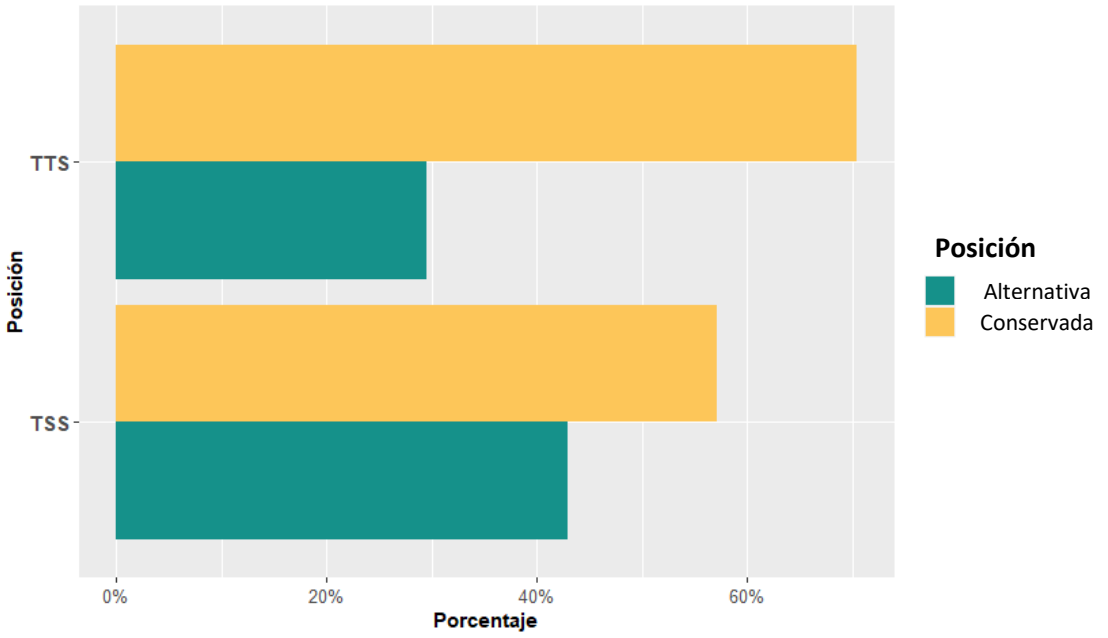

Figura 13. Características de los sitios de inicio y final de las isoformas en los distintos genes. Distribución en porcentaje de aquellos genes cuyos transcritos terminan (TTS) o empiezan (TSS) todos igual (Posición conservada) o tienen algún transcrito con alguno de estos sitios diferente (Posición alternativa).

4.3.5. Análisis de términos GO.

Se realizó un análisis de los términos GO presentes en los genes que presentaban diferentes tipos de splicing (Figura 14A). El objetivo de este análisis es ver si con los resultados del *pipeline* se pueden extraer conclusiones biológicas relevantes a nivel genómico. Se observó una sobrerrepresentación de términos GO relacionados con la unión proteica (con IDs GO:0005515, GO:0005488) y las funciones dependientes de ATP (GO:0140657) en los genes que presentan cambios exónicos. De la misma manera, se detectó una sobrerrepresentación de las funciones relacionadas con las respuestas de defensa en los genes que presentan retenciones intrónicas. Los genes con A3 o A5 no presentaron ningún término GO relevante.

También se analizaron los términos GO presentes en los genes con sitios de inicio y final de transcrito alternativos, obteniendo el mismo resultado que en los cambios exónicos a excepción de las funciones dependientes de ATP.

Por último, debido a la anomalía detectada en la longitud de los transcritos no codificantes antes comentada (Figrua 10C), se analizaron los términos GO presentes en los mismos. Obteniendo una gran cantidad de términos GO relacionados con la cadena de transporte de electrones y la respiración celular (Figura14B).

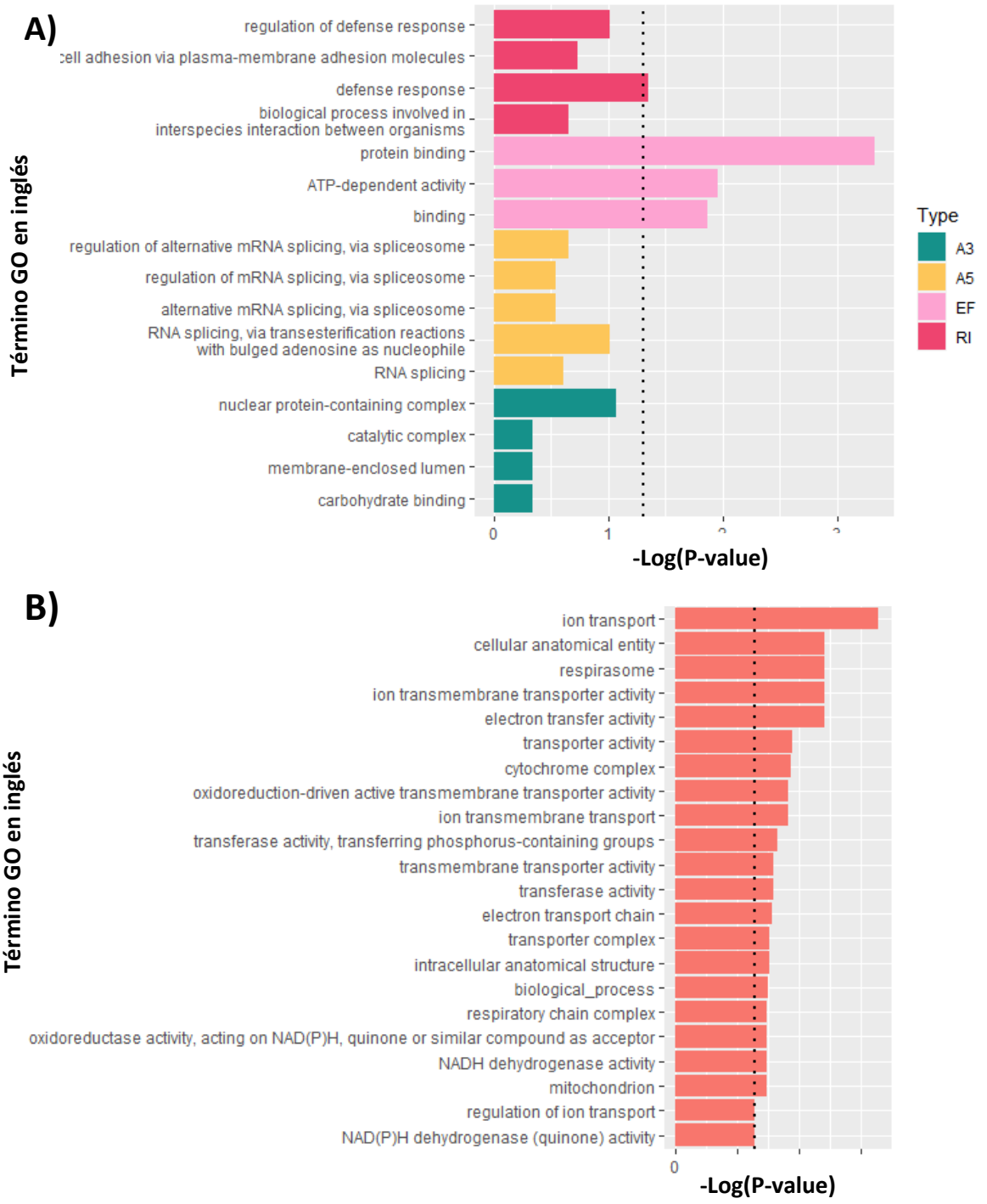

Figura 14 Análisis de términos GO de los diferentes tipos de splicing. A) Resultados del análisis de términos GO sobrerrepresentados en los distintos tipos de splicing, aquellos estadísticamente significativos son los que superan el tresshold de 0,05 de p-value (1,33 en -Log(P-value)). Los términos GO que se han analizado pertenecen a todos los niveles de la nomenclatura, desde el 1 hasta el 15. **B)** Resultados del análisis de términos GO sobrerrepresentados en los genes con transcritos no codificantes

5. DISCUSIÓN

5.1. OPTIMIZACIÓN DEL *PIPELINE*.

Uno de los problemas principales a los que tiene que hacer frente la bioinformática es el alto coste computacional de las herramientas que se desarrollan. Un algoritmo poco eficiente que funciona bien con pocos datos puede llevar a varias horas o incluso días cuando se utiliza con los datos reales, debido a la naturaleza masiva de los datos analizados, sobre todo los genómicos y transcriptómicos (Hanussek et al., 2021). Por ello, es muy importante revisar y optimizar las primeras versiones de cualquier tipo de software bioinformático antes de escalarlo para su uso real. Tal y como muestran los resultados, se ha mejorado sensiblemente la velocidad a la que trabaja el *pipeline*. Existe un cuello de botella en la primera parte de este, pues COGENT, la parte más demandante computacionalmente, es un programa externo y por tanto no se puede optimizar. Sin embargo, se ha conseguido aliviar en parte acelerando significativamente la parte final del *pipeline*, haciendo su uso más accesible para los posibles usuarios.

Por otra parte, otro de los problemas comunes entre estos programas son los errores en el código que obligan a hacer actualizaciones constantes corrigiéndolos. Durante la optimización se han corregido los errores encontrados en las diferentes funciones de clasificación, estos errores comprometían aspectos del *pipeline* que no se incluyeron en la validación por ser accesorios, por lo que de no haber sido revisados habrían ocasionado problemas en el futuro. Un ejemplo es el error al asignar la región del transcrito en la que se encuentran las retenciones intrónicas, lo cual podría llevar a conclusiones biológicas erróneas. También se ha solucionado la incorrecta clasificación de eventos de retención intrónica como inclusiones exónicas, problema que podría haber alterado los resultados en por ejemplo un análisis de términos GO. Además, esto es especialmente problemático no solo a nivel transcriptómico, sino también a nivel del estudio de una sola isoforma ya que para un usuario podría ser difícil distinguir una inclusión exónica de una retención intrónica utilizando las herramientas de visualización disponibles como SplicingViewer.

Por último, el cambio en el tipo de mapeo de cada transcrito (antes contra todo el genoma reconstruido y ahora tan solo contra la secuencia del locus del que proviene) ahorra problemas con el código (debido a esto se generaban errores bajo circunstancias concretas que hacían que el *pipeline* se parase y no finalizase en análisis) y además evita una incorrecta detección de los eventos de splicing. Los transcritos que mapeaban en locus incorrectos no lo hacían completamente, por lo que no se detectaban todos sus fragmentos, perdiendo información muy importante para la posterior detección. Los motivos por los cuales estos transcritos mapeaban a otros locus son varios; en ocasiones estos dos locus eran muy similares. La separación de los transcritos en diferentes familias génicas se realiza mediante MASH (Ondov et al., 2016), un programa que utiliza una estrategia basada en K-mers para deducir la similitud, mientras que el mapeo para detectar los fragmentos se realiza con minimap2 (H. Li, 2018), que utiliza un algoritmo diferente. Por tanto, cabe la posibilidad de que MASH esté separando en 2 familias génicas los transcritos que en realidad pertenecen a un solo gen. Sin embargo, también se puede dar el caso de que MASH esté separando correctamente los transcritos, pero minimap2 no sea tan sensible como para detectar esta diferencia. Es probable que la realidad sea una mezcla de ambas posibilidades.

5.2. VALIDACIÓN DEL *PIPELINE*.

En cuanto a la validación de la precisión del *pipeline* a la hora de detectar los diferentes eventos de splicing y anotarlos en los fragmentos detectados podemos decir que los resultados son dispares.

Si nos fijamos en el promedio de todas las categorías, vemos un buen resultado en cuanto al ratio de falsos positivos, lo cual indica que cuando un evento es clasificado, esa clasificación pocas veces es incorrecta. Esto se ve reflejado en la especificidad que es muy alta. Sin embargo, la ratio de falsos negativos si es bastante alto, lo que indica que muchos de los eventos reales no están siendo detectados, lo que se refleja en una baja sensibilidad.

En cualquier caso, los números obtenidos para cada categoría difieren entre sí. La sensibilidad del *pipeline* para detectar A5 y A3 es muy baja en ambas especies, pues como vemos en la matriz de confusión, estos eventos son confundidos con fragmentos exónicos en la mayoría de los casos. La predicción de retenciones intrónicas y los fragmentos exónicos presentan diferencias en las dos especies. La predicción de las retenciones intrónicas ha sido prácticamente igual de específica en ambas especies, pero ha sido más sensible en Drosophila que en Pez cebra. Esto implica que muchas retenciones intrónicas han sido confundidas con fragmentos exónicos por el *pipeline* en esta última especie, al igual que ocurría con los A5 y A3. Estas confusiones son la causa de la baja especificidad que encontramos en la detección de fragmentos exónicos, sobre todo en pez cebra (hay muchos fragmentos detectados como fragmentos exónicos que en realidad son otro evento de splicing). Por contraparte la detección de fragmentos exónicos destaca muy por encima del resto de categorías en cuanto a sensibilidad.

En general, podemos decir que los valores obtenidos no son óptimos, pues la sensibilidad promedio es menor al 50%. Además, los valores obtenidos para la especificidad no son demasiado fiables, ya que están influenciados por el alto número de exones. Al utilizar datos reales nos encontramos con la situación de que las diferentes categorías no están equitativamente representadas, haciendo que haya un número muy superior de fragmentos exónicos reales. Esto hace que el número de negativos reales (verdaderos negativos+falsos positivos) aumente para el resto de las categorías, y por tanto el ratio de falsos positivos (falsos positivos dividido entre negativos reales) sea muy bajo siempre.

Los resultados parecen indicar que la detección de splicing alternativo con este método es muy dependiente de las características biológicas del mismo. Por ejemplo, vemos una diferencia entre la detección de retenciones intrónicas en pez cebra y Drosophila, y si nos fijamos en las características de sus intrónes vemos como hay una diferencia de tamaño de casi 1500 pares de bases (Haddrill et al., 2005; Moss et al., 2011). También encontramos diferencias similares entre A5/A3 y el resto de eventos.

Pequeñas diferencias en los transcritos pueden alterar mucho la secuencia reconstruida de COGENT. SUPPA utiliza las posiciones genómicas para la detección, por lo que va a ser muy sensible a pequeños errores en la reconstrucción. Para ilustrar esto tenemos el ejemplo de la pobre detección de A5, estos eventos son en ocasiones muy pequeños (incluso por debajo de 50 nucleótidos) (Akerman & Mandel-Gutfreund, 2007; Xia et al., 2006). Durante la reconstrucción del locus de referencia una diferencia muy pequeña entre 2 transcritos genera ambigüedades en los grafos de bruijn, por lo que puede llegar a ser ignorada por COGENT, no incluyéndola en la secuencia. En otras ocasiones, si no se consigue resolver la ambigüedad el locus puede acabar dividido en 2, comprometiendo también la detección.

En definitiva, la complejidad del splicing genera problemas en las reconstrucciones de COGENT, lo que se propaga y amplifica al ser estas reconstrucciones la base de la metodología elegida para la detección de splicing. Quizás, sería conveniente probar otras herramientas diferentes a SUPPA, no basadas en el mapeo de secuencias o menos dependiente de esto para la detección. Algunas alternativas serían el uso de predictores de splicing basados en la información de la secuencia (presencia de secuencias de

splicing canónicas, enhancers, etc) como los programas descritos en Jaganathan et al., 2019 y Paggi & Bejerano, 2018. Por otra parte, es importante destacar que la mayoría de eventos de splicing SI se están detectando, sin embargo no se consigue distinguir correctamente a que categoría pertenece cada evento. Por tanto, otra posibilidad para mejorar los resultados obtenidos es tener en cuenta solo la posición en la que encontramos el evento y utilizar herramientas como el aprendizaje automático para discernir de que tipo de evento se trata.

5.3. APLICACIÓN A UN CASO REAL.

Se analizó el transcriptoma de Micropterus salmoides para comprobar la funcionalidad y utilidad de la metodología seleccionada. Al ser la cantidad de datos mayor, este enfoque nos permite hacer un análisis más extensivo de la actuación tanto individual como colectiva de las diferentes herramientas que componen el *pipeline*.

Reconstrucción de Cogent

Respecto a la actuación de COGENT vemos como el programa ha sido capaz de agrupar gran parte de los diferentes transcritos en 7322 familias génicas y reconstruir la secuencia codificante de todas ellas. Sin embargo, hay un considerable número de transcritos que no se han conseguido agrupar (un 36%) y han sido descartados por el software. Esta pérdida de transcritos puede comprometer la detección de los eventos de splicing, ya sea porque contengan algún evento o porque la falta de su secuencia pueda comprometer la reconstrucción. El número de familias génicas descubiertas (7322) es inferior al que encontramos para otras especies cercanas como *Dicentrarchus labrax* (26,719 genes) (Tine et al., 2014). Sin embargo, esto puede tener una explicación biológica: El número de genes expresados es inferior a la totalidad del genoma. Además, los genes homólogos o pertenecientes a la misma familia génica son indistinguibles por la metodología de COGENT debido a su elevada identidad) identidad entre ambos (>80%) (Y. Zhang et al., 2020). Si a estas dos situaciones le sumamos la pérdida de transcritos antes comentada se puede explicar el menor número de genes obtenidos.

Teóricamente, deberíamos obtener un locus reconstruido para cada familia génica, sin embargo, esto no ha sido así, encontrándonos génes con desde 2 hasta 21 reconstrucciones diferentes. Para reconstruir la secuencia se dividen los transcritos en K-mers para posteriormente intentar ordenar en unos grafos de Bruijn. Esto hace que en ocasiones y por diferentes motivos el programa no pueda resolver ambigüedades y no sea capaz de obtener una secuencia completa, resultando en varias reconstrucciones inconexas (en inglés “contigs”). Esto último es especialmente relevante para la posterior predicción de eventos de splicing ya que frecuentemente estas ambigüedades se deben precisamente a eventos A5 o A3; otra posible explicación es la presencia de transcritos de diferentes genes en el mismo cluster (debido a las familias génicas y genes homólogos antes comentados). Estas situaciones pueden resultar en predicciones incorrectas.

Anotación de fragmentos y clasificación de los transcritos.

La media de fragmentos exónicos detectados es de 5,395, casi la mitad de lo encontrado en otros peces actinopterigios como el pez damisela (*Acanthochromis polyacanthus*) con 9.5 exones por gen (*Acanthochromis Polyacanthus Annotation Report*, n.d.) . Estas diferencias con la bibliografía se pueden explicar por la forma de detectar los fragmentos exónicos por parte del *pipeline*. Tal y como se ha explicado anteriormente, dos exones que no sufren ninguna alteración en las isoformas del gen nunca se podrán detectar como 2 fragmentos independientes, si no como un solo fragmento exónico. Por tanto, los fragmentos exónicos que detectamos no se corresponden con un solo exón. Además, muchas isoformas son específicas de tejido y a pesar de partir de una mezcla de tejidos representativos no son

la totalidad de estos. Esto implica que habrá isoformas imposibles de detectar, y por tanto exones cuya secuencia no podrá nunca estar incluida en las reconstrucciones.

También observamos un menor número (3,85) de genes con 1 solo fragmento exónico respecto a otras especies similares como la lubina europea (*Dicentrarchus labrax)* que presenta un 8,3% (Tine et al., 2021). Los genes de un solo exón presentan funcionalidad, y hay interés por su conocimiento por lo que es importante detectarlos correctamente. Como ya hemos comentado, debido a la naturaleza de la detección de fragmentos en el *pipeline* no podemos asegurarnos de que se trate realmente de genes con un solo exón.

En general, vemos que es importante tener en cuenta que los fragmentos exónicos NO son equivalentes a exones, y por tanto hay que tener esto en cuenta antes de extraer conclusiones biológicas sobre estos.

Respecto a la detección de eventos de splicing y la clasificación de los transcritos vemos que el evento más común son los cambios en fragmentos exónicos. La distribución de los eventos de splicing es muy dependiente de la especie y en general los cambios exónicos son, por mucho, el evento más común en animales. Esto se ajusta a los resultados obtenidos, al igual que el porcentaje obtenido de retención intrónica (en torno al 5-6%), sin embargo, A5 y A3 deberían haberse detectado en una mayor cantidad, pues constituyen el segundo evento más común en la bibliografía (Chaudhary et al., 2019). El bajo porcentaje que representan se debe principalmente a la baja sensibilidad del *pipeline* para detectar estos eventos, una baja sensibilidad implica necesariamente la no detección de gran parte de estos eventos.

El *pipeline* ha sido capaz de detectar cambios en la región transcriptómica en la que ocurren los eventos de splicing. Normalmente, los eventos de splicing que afectan a la UTR suelen estar más relacionados con funciones regulatorias afectando por ejemplo a la estabilidad del transcrito (Thiele et al., 2006) o a la eficiencia de reconocimiento por parte del ribosoma (Tamarkin-Ben-Harush et al., 2017), mientras que cambios en la zona codificante suele dar lugar a cambios en la estructura de la proteína. Los cambios en fragmentos exónicos son más comunes en las UTRs, esto puede ser debido a que dentro de los cambios exónicos se incluyen la poliadenilacion alternativa y el uso de promotor alternativo, eventos de splicing relativamente comunes y que solo pueden afectar a exones que se encuentran en las UTR (el primero y el último exón) (Chang et al., 2015; Huin et al., 2017; Y. Zhang, Liu, et al., 2021). Por su parte destaca la presencia de retenciones intrónicas en la zona codificante (CDS), lo cual alinea con lo observado en animales debiéndose a una mayor presencia de intrones en estas zonas (Galante et al., 2004). Para A3 y A5 no se ha encontrado ninguna diferencia notable, lo cual podría ser debido a la pobre detección de estos.

Potencial codificante.

GeneMarkTS ha identificado en torno a un 10% de transcritos no codificantes, los cuales tenían una distribución de tamaño peculiar por no seguir una distribución normal. Existe un pico de transcritos predichos como no codificantes que supera por mucho la longitud típica de los ARN no codificantes más largos para esta misma especie (Zhu et al., 2022). Además, este pico coincide con el pico de ARN mensajeros codificantes, por lo que podría tratarse de algún error.

Se realizó un análisis de los términos GO presentes en estos transcritos no codificantes, el cual no debería arrojar ningún resultado significativo al no codificar para proteínas (Blast2GO tiene una herramienta especial para anotar su funcionalidad, pero en este caso no fue utilizada). Sin embargo, se encontraron una gran cantidad de términos GO sobrerrepresentados en estos transcritos, todos relacionado con la cadena de transporte de electrones. Es cierto que algunos ARN no codificantes pueden obtener anotaciones en Blast2GO por guardar identidad con otros genes codificantes o por codificar pequeños péptidos (Xing et al., 2021) , sin embargo, en la bibliografía no se ha encontrado una relación significativa entre los ARN no codificantes y los procesos de respiración celular con anterioridad. Es posible que parte de estos ARN no codificantes se correspondan con mensajeros trucados durante el

proceso de secuenciación, provocando que les falten las secuencias necesarias en la 5’ UTR para ser predichos como codificantes.

Términos GO asociados.

Además de lo antes comentado se han encontrado términos GO enriquecidos en aquellos genes que presentan cambios exónicos respecto al resto de genes. Estos términos GO estaban todos relacionados con funciones de unión a proteínas, sin embargo, no se han encontrado resultados similares en la bibliografía. También se han encontrado una sobrerrepresentación de las respuestas defensivas en los genes que presentan retenciones intrónicas, encontramos ejemplos de la implicación de la retención intrónica con la regulación inmune, como por ejemplo la diferenciación de células inmunes (Song et al., 2022). En los eventos A3/5 no se ha encontrado ningún término GO asociado, lo que de nuevo puede ser un síntoma de la poca sensibilidad del pipeline para detectarlos.

Los resultados obtenidos en la aplicación del *pipeline* sobre datos reales ponen de manifiesto el gran potencial de las herramientas disponibles en la actualidad para el análisis transcriptómico de especies no modelo. Gracias a utilizar varias de estas herramientas en conjunto se puede realizar una gran variedad de análisis, permitiéndonos plantear y responder las mismas preguntas biológicas que cuando si tenemos un genoma de referencia de calidad: Potencial codificante, composición genómica y transcriptómica, análisis funcionales, detección de eventos de splicing, etc. Sin embargo, también ponen de manifiesto que estas herramientas solo nos proporcionan una visión aproximada de la realidad biológica debido a las limitaciones derivadas de no disponer de un genoma de referencia.

Ejemplos de esta limitación son la imposibilidad de equiparar los fragmentos exónicos detectados por COGENT a los exones reales, la incorrecta agrupación de algunos transcritos debido a homologías, así como la imposibilidad de reconstruir una secuencia codificante sin errores en genes con mucho splicing. Esto último impide el buen funcionamiento de SUPPA, uno de los detectores de splicing más utilizados, pero que tiene limitaciones para detectar correctamente los eventos cuando no tenemos la información del genoma.

6. CONCLUSIÓN

Los resultados obtenidos y discutidos en este trabajo nos permiten sacar una serie de conclusiones:

En primer lugar, el *pipeline* ha sido optimizado en cuanto a errores y velocidad, lo cual permite un uso más versátil del mismo por parte de los usuarios, evitando además la necesidad de corregir algunos errores en el futuro. Sin embargo, sigue existiendo un cuello de botella en cuanto a la velocidad, ya que COGENT es una herramienta muy demandante computacionalmente.

En segundo lugar, la metodología utilizada para detectar los eventos de splicing, aunque muestra alta especificidad, necesita ser revisada para mejorar su sensibilidad. Los pequeños cambios introducidos por COGENT a la hora de reconstruir las secuencias del locus se propagan conforme avanza el *pipeline* debido a la dependencia de todos los pasos de estas secuencias reconstruidas. Estos pequeños errores se traducen en dificultad para detectar sobre todo los eventos A3 y A5, aunque también influyen en la detección de las retenciones intrónicas. Sin embargo, si no tenemos en cuenta el tipo de evento de splicing el *pipeline* si funciona correctamente a nivel general, lo cual puede constituir un punto de partida para mejorar los resultados obtenidos introduciendo cambios en la metodología.

En tercer lugar, la aplicación del *pipeline* a un caso real demuestra la utilidad de las herramientas incluidas, así como la versatilidad que proporciona para realizar diferentes análisis: Detección de splicing alternativo, detección del potencial codificante de los transcritos (como primer paso para la detección de ARN no codificantes de distintos tipos), análisis de términos GO, etc. Sin embargo, hay que ser conscientes de las limitaciones de estas herramientas en la interpretación de resultados. La baja fiabilidad para la detección de algunos de los eventos compromete los resultados finales como se ha podido comprobar aquí. A la hora de sacar conclusiones relacionados con el número de exones hay que tener en cuenta que se detectan fragmentos exónicos y no exones, por lo que no pueden equipararse. Por su parte el análisis del potencial codificante debe ser tomado con precaución debido a las limitaciones de la tecnología de secuenciación por lectura larga.

En líneas generales los resultados evidencian la complejidad del análisis transcriptómico, en especial en el análisis del splicing alternativo cuando no tenemos un genoma de referencia de calidad. Las herramientas disponibles nos permiten aproximarnos, pero no se llega a los estándares alcanzados en los estudios cuando si se dispone de un genoma. El nivel de desarrollo actual puede ser un gran punto de inicio, pues disponemos de diversas herramientas con un gran potencial, sin embargo, se necesita un avance en el desarrollo de algoritmos y software que permita un análisis más exacto del transcriptoma de este tipo de especies.

Mientras tanto, los resultados obtenidos en aquellos experimentos que utilizen estas herramientas deben ser tomados con cautela, y toda conclusión debe ser debidamente validada con algún otro método de referencia.

7. BIBLIOGRAFÍA.

Acanthochromis polyacanthus Annotation Report. (n.d.). Retrieved July 28, 2022, from https://www.ncbi.nlm.nih.gov/genome/annotation_euk/Acanthochromis_polyacanthus/100/

Akerman, M., & Mandel-Gutfreund, Y. (2007). Does distance matter? Variations in alternative 3' splicing regulation. *Nucleic Acids Research*, *35*(16), 5487. https://doi.org/10.1093/NAR/GKM603

Alamancos, G. P., Pagès, A., Trincado, J. L., Bellora, N., & Eyras, E. (2015a). Leveraging transcript quantification for fast computation of alternative splicing profiles. *RNA*, *21*(9), 1521–1531. https://doi.org/10.1261/RNA.051557.115

Alden, K., & Read, M. (2013). Computing: Scientific software needs quality control. *Nature*, *502*(7472), 448. https://doi.org/10.1038/502448D

Bayega, A., Oikonomopoulos, S., Gregoriou, M. E., Tsoumani, K. T., Giakountis, A., Wang, Y. C., Mathiopoulos, K. D., & Ragoussis, J. (2021). Nanopore long-read RNA-seq and absolute quantification delineate transcription dynamics in early embryo development of an insect pest. *Scientific Reports 2021 11:1*, *11*(1), 1–14. https://doi.org/10.1038/s41598-021-86753-7

Benesova, S., Kubista, M., & Valihrach, L. (2021). Small RNA-Sequencing: Approaches and Considerations for miRNA Analysis. *Diagnostics 2021, Vol. 11, Page 964*, *11*(6), 964. https://doi.org/10.3390/DIAGNOSTICS11060964

Berget, S. M., Moore, C., & Sharp, P. A. (1977). Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences of the United States of America*, *74*(8), 3171–3175. https://doi.org/10.1073/PNAS.74.8.3171

Besemer, J., Lomsadze, A., & Borodovsky, M. (2001). GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, *29*(12), 2607–2618. https://doi.org/10.1093/NAR/29.12.2607

Chaisson, M. J. P., Wilson, R. K., & Eichler, E. E. (2015). Genetic variation and the de novo assembly of human genomes. *Nature Reviews Genetics 2015 16:11*, *16*(11), 627–640. https://doi.org/10.1038/nrg3933

Chang, J. W., Zhang, W., Yeh, H. S., de Jong, E. P., Jun, S., Kim, K. H., Bae, S. S., Beckman, K., Hwang, T. H., Kim, K. S., Kim, D. H., Griffin, T. J., Kuang, R., & Yong, J. (2015). mRNA 3'-UTR shortening is a molecular signature of mTORC1 activation. *Nature Communications 2015 6:1*, *6*(1), 1–9. https://doi.org/10.1038/ncomms8218

Chaudhary, S., Khokhar, W., Jabre, I., Reddy, A. S. N., Byrne, L. J., Wilson, C. M., & Syed, N. H. (2019). Alternative splicing and protein diversity: Plants versus animals. *Frontiers in Plant Science*, *10*, 708. https://doi.org/10.3389/FPLS.2019.00708/BIBTEX

Claverie, J.-M. (2001). What If There Are Only 30,000 Human Genes? *Science*, *291*(5507), 1255–1257. https://doi.org/10.1126/science.1058969

Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., & Robles, M. (2005). Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, *21*(18), 3674–3676. https://doi.org/10.1093/BIOINFORMATICS/BTI610

Consortium*, T. C. elegans S. (1998). Genome sequence of the nematode C. elegans: A platform for investigating biology. *Science*, *282*(5396), 2012–2018. https://doi.org/10.1126/SCIENCE.282.5396.2012/SUPPL_FILE/C-ELEGANS.XHTML

Craig Venter, J., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., … Zhu, X. (2001). The sequence of the human genome. *Science*, *291*(5507), 1304–1351. https://doi.org/10.1126/SCIENCE.1058040/SUPPL_FILE/C18_SCIENCE.PDF

Feng, D., & Xie, J. (2013). Aberrant splicing in neurological diseases. *Wiley Interdisciplinary Reviews. RNA*, *4*(6), 631–649. https://doi.org/10.1002/WRNA.1184

Feng, H., Qin, Z., & Zhang, X. (2013). Opportunities and methods for studying alternative splicing in cancer with RNA-Seq. *Cancer Letters*, *340*(2), 179–191. https://doi.org/10.1016/J.CANLET.2012.11.010

Galante, P. A. F., Sakabe, N. J., Kirschbaum-Slager, N., & de Souza, S. J. (2004). Detection and evaluation of intron retention events in the human transcriptome. *RNA*, *10*(5), 757. https://doi.org/10.1261/RNA.5123504

Giannoulatou, E., Park, S. H., Humphreys, D. T., & Ho, J. W. K. (2014). Verification and validation of bioinformatics software without a gold standard: a case study of BWA and Bowtie. *BMC Bioinformatics*, *15*(Suppl 16), S15. https://doi.org/10.1186/1471-2105-15-S16-S15

Haddrill, P. R., Charlesworth, B., Halligan, D. L., & Andolfatto, P. (2005). Patterns of intron sequence evolution in Drosophila are dependent upon length and GC content. *Genome Biology*, *6*(8), R67. https://doi.org/10.1186/GB-2005-6-8-R67

Halperin, R. F., Hegde, A., Lang, J. D., Raupach, E. A., Narayanan, V., Huentelman, M., Belnap, N., Aziz, A. M., Ramsey, K., Legendre, C., Liang, W. S., LoRusso, P. M., Sekulic, A., Sosman, J. A., Trent, J. M., Rangasamy, S., Pirrotte, P., & Schork, N. J. (2021). Improved methods for RNAseq-based alternative splicing analysis. *Scientific Reports 2021 11:1*, *11*(1), 1–15. https://doi.org/10.1038/s41598-021-89938-2

Hanussek, M., Bartusch, F., & Ger, J. K. (2021). Performance and scaling behavior of bioinformatic applications in virtualization environments to create awareness for the efficient use of compute resources. *PLOS Computational Biology*, *17*(7), e1009244. https://doi.org/10.1371/JOURNAL.PCBI.1009244

Huin, V., Buée, L., Behal, H., Labreuche, J., Sablonnière, B., & Dhaenens, C. M. (2017). Alternative promoter usage generates novel shorter MAPT mRNA transcripts in Alzheimer's disease and progressive supranuclear palsy brains. *Scientific Reports 2017 7:1*, *7*(1), 1–10. https://doi.org/10.1038/s41598-017-12955-7

Jaganathan, K., Kyriazopoulou Panagiotopoulou, S., McRae, J. F., Darbandi, S. F., Knowles, D., Li, Y. I., Kosmicki, J. A., Arbelaez, J., Cui, W., Schwartz, G. B., Chow, E. D., Kanterakis, E., Gao, H., Kia, A., Batzoglou, S., Sanders, S. J., & Farh, K. K. H. (2019). Predicting Splicing from Primary Sequence with Deep Learning. *Cell*, *176*(3), 535-548.e24. https://doi.org/10.1016/J.CELL.2018.12.015

Jiang, Z., Zhong, Z., Miao, Q., Zhang, Y., Ni, B., Zhang, M., & Tang, J. (2021). circPTPN22 as a novel biomarker and ceRNA in peripheral blood mononuclear cells of rheumatoid arthritis. *Molecular Medicine Reports*, *24*(2), 1–11. https://doi.org/10.3892/MMR.2021.12256/HTML

Kamali, A. H., Giannoulatou, E., Chen, T. Y., Charleston, M. A., McEwan, A. L., & Ho, J. W. K. (2015a). How to test bioinformatics software? *Biophysical Reviews*, *7*(3), 343. https://doi.org/10.1007/S12551-015-0177-3

Kaul, S., Koo, H. L., Jenkins, J., Rizzo, M., Rooney, T., Tallon, L. J., Feldblyum, T., Nierman, W., Benito, M. I., Lin, X., Town, C. D., Venter, J. C., Fraser, C. M., Tabata, S., Nakamura, Y., Kaneko, T., Sato, S., Asamizu, E., Kato, T., … Somerville, C. (2000). Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature 2000 408:6814*, *408*(6814), 796–815. https://doi.org/10.1038/35048692

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., Fitzhugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., Levine, R., McEwan, P., … Morgan, M. J. (2001). Initial sequencing and analysis of the human genome. *Nature*, *409*(6822), 860–921. https://doi.org/10.1038/35057062

Lawrence, M., Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan, M. T., & Carey, V. J. (2013). Software for Computing and Annotating Genomic Ranges. *PLOS Computational Biology*, *9*(8), e1003118. https://doi.org/10.1371/JOURNAL.PCBI.1003118

Lei, H., & Vořechovský, I. (2005). Identification of Splicing Silencers and Enhancers in Sense Alus: a Role for Pseudoacceptors in Splice Site Repression. *Molecular and Cellular Biology*, *25*(16), 6912. https://doi.org/10.1128/MCB.25.16.6912-6920.2005

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, *34*(18), 3094–3100. https://doi.org/10.1093/BIOINFORMATICS/BTY191

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, *25*(14), 1754–1760. https://doi.org/10.1093/BIOINFORMATICS/BTP324

Li, T., Hu, D., & Gong, Y. (2021). Identification of potential lncRNAs and co-expressed mRNAs in gestational diabetes mellitus by RNA sequencing. *Journal of Maternal-Fetal and Neonatal Medicine*. https://doi.org/10.1080/14767058.2021.1875432

Liu, X., Mei, W., Soltis, P. S., Soltis, D. E., & Barbazuk, W. B. (2017). Detecting alternatively spliced transcript isoforms from single-molecule long-read sequences without a reference genome. *Molecular Ecology Resources*, *17*(6), 1243–1256. https://doi.org/10.1111/1755-0998.12670

Liu, Y., Gonzàlez-Porta, M., Santos, S., Brazma, A., Marioni, J. C., Aebersold, R., Venkitaraman, A. R., & Wickramasinghe, V. O. (2017). Impact of Alternative Splicing on the Human Proteome. *Cell Reports*, *20*(5), 1229. https://doi.org/10.1016/J.CELREP.2017.07.025

Louadi, Z., Oubounyt, M., Tayara, H., & To Chong, K. (2019). Deep Splicing Code: Classifying Alternative Splicing Events Using Deep Learning. *Genes 2019, Vol. 10, Page 587*, *10*(8), 587. https://doi.org/10.3390/GENES10080587

Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. *PLoS Computational Biology*, *13*(5). https://doi.org/10.1371/JOURNAL.PCBI.1005457

Manzoni, C., Kia, D. A., Vandrovcova, J., Hardy, J., Wood, N. W., Lewis, P. A., & Ferrari, R. (2018). Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences. *Briefings in Bioinformatics*, *19*(2), 286–302. https://doi.org/10.1093/BIB/BBW114

Marra, M. A., Hillier, L., & Waterston, R. H. (1998). Expressed sequence tags--ESTablishing bridges between genomes. *Trends in Genetics : TIG*, *14*(1), 4–7. https://doi.org/10.1016/S0168-9525(97)01355-3

Mayeda' And, A., & Ohshima2, Y. (1988). Short donor site sequences inserted within the intron of beta-globin pre-mRNA serve for splicing in vitro. *Molecular and Cellular Biology*, *8*(10), 4484–4491. https://doi.org/10.1128/MCB.8.10.4484-4491.1988

McGettigan, P. A. (2013). Transcriptomics in the RNA-seq era. *Current Opinion in Chemical Biology*, *17*(1), 4–11. https://doi.org/10.1016/J.CBPA.2012.12.008

Min, F., Wang, S., & Zhang, L. (2015). Survey of Programs Used to Detect Alternative Splicing Isoforms from Deep Sequencing Data in Silico. *BioMed Research International*, *2015*. https://doi.org/10.1155/2015/831352

Mortazavi, A., Williams, B. A., Mccue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature.Com*, *5*(7), 621. https://doi.org/10.1038/NMETH.1226

Moss, S. P., Joyce, D. A., Humphries, S., Tindall, K. J., & Lunt, D. H. (2011). Comparative Analysis of Teleost Genome Sequences Reveals an Ancient Intron Size Expansion in the Zebrafish Lineage. *Genome Biology and Evolution*, *3*(1), 1187. https://doi.org/10.1093/GBE/EVR090

Muller, I. B., Meijers, S., Kampstra, P., van Dijk, S., van Elswijk, M., Lin, M., Wojtuszkiewicz, A. M., Jansen, G., de Jonge, R., & Cloos, J. (2021). Computational comparison of common event-based differential splicing tools: practical considerations for laboratory researchers. *BMC Bioinformatics*, *22*(1), 347. https://doi.org/10.1186/S12859-021-04263-9/TABLES/2

Nagoshi, R. N., & Baker, B. S. (1990). Regulation of sex-specific RNA splicing at the Drosophila doublesex gene: cis-acting mutations in exon sequences alter sex-specific RNA splicing patterns. *Genes & Development*, *4*(1), 89–97. https://doi.org/10.1101/GAD.4.1.89

Neff, R. A., Wang, M., Vatansever, S., Guo, L., Ming, C., Wang, Q., Wang, E., Horgusluoglu-Moloch, E., Song, W. M., Li, A., Castranio, E. L., Julia, T. C. W., Ho, L., Goate, A., Fossati, V., Noggle, S., Gandy, S., Ehrlich, M. E., Katsel, P., … Zhang, B. (2021). Molecular subtyping of Alzheimer's disease using RNA sequencing data reveals novel mechanisms and targets. *Science Advances*, *7*(2). https://doi.org/10.1126/SCIADV.ABB5398/SUPPL_FILE/ABB5398_SM.PDF

Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., & Phillippy, A. M. (2016). Mash: Fast genome and metagenome distance estimation using MinHash. *Genome Biology*, *17*(1), 1–14. https://doi.org/10.1186/S13059-016-0997-X/FIGURES/5

O'Rawe, J., Jiang, T., Sun, G., Wu, Y., Wang, W., Hu, J., Bodily, P., Tian, L., Hakonarson, H., Johnson, W. E., Wei, Z., Wang, K., & Lyon, G. J. (2013). Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome Medicine*, *5*(3). https://doi.org/10.1186/GM432

Paggi, J. M., & Bejerano, G. (2018). A sequence-based, deep learning model accurately predicts RNA splicing branchpoints. *RNA*, *24*(12), 1647–1653. https://doi.org/10.1261/RNA.066290.118/-/DC1

Park, E., Pan, Z., Zhang, Z., Lin, L., & Xing, Y. (2018). The Expanding Landscape of Alternative Splicing Variation in Human Populations. *The American Journal of Human Genetics*, *102*(1), 11–26. https://doi.org/10.1016/J.AJHG.2017.11.002

Pennisi, E. (2000). And the gene number is ...? *Science*, *288*(5469), 1146–1147. https://doi.org/10.1126/SCIENCE.288.5469.1146/ASSET/CF58B99D-AE3E-4595-82D3-84DA6EA44B99/ASSETS/GRAPHIC/1146-1.GIF

Pinkney, H. R., Black, M. A., & Diermeier, S. D. (2021). Single-Cell RNA-Seq Reveals Heterogeneous lncRNA Expression in Xenografted Triple-Negative Breast Cancer Cells. *Biology 2021, Vol. 10, Page 987*, *10*(10), 987. https://doi.org/10.3390/BIOLOGY10100987

Poplin, R., Chang, P. C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P. T., Gross, S. S., Dorfman, L., McLean, C. Y., & Depristo, M. A. (2018). A universal SNP and small-indel variant caller using deep neural networks. *Nature Biotechnology 2018 36:10*, *36*(10), 983–987. https://doi.org/10.1038/nbt.4235

Riepe, T. v., Khan, M., Roosing, S., Cremers, F. P. M., & 't Hoen, P. A. C. (2021). Benchmarking deep learning splice prediction tools using functional splice assays. *Human Mutation*, *42*(7), 799–810. https://doi.org/10.1002/HUMU.24212

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, *270*(5235), 467–470. https://doi.org/10.1126/SCIENCE.270.5235.467

Sims, D., Sudbery, I., Ilott, N. E., Heger, A., & Ponting, C. P. (2014). *Sequencing depth and coverage: key considerations in genomic analyses*. https://doi.org/10.1038/nrg3642

Song, R., Tikoo, S., Jain, R., Pinello, N., Au, A. Y. M., Nagarajah, R., Porse, B., Rasko, J. E. J., & J.-L. Wong, J. (2022). Dynamic intron retention modulates gene expression in the monocytic differentiation pathway. *Immunology*, *165*(2), 274–286. https://doi.org/10.1111/IMM.13435

Tabuchi, K., & Südhof, T. C. (2002). Structure and evolution of neurexin genes: Insight into the mechanism of alternative splicing. *Genomics*, *79*(6), 849–859. https://doi.org/10.1006/GENO.2002.6780

Tamarkin-Ben-Harush, A., Vasseur, J.-J., oise Debart, F., Ulitsky, I., & Dikstein, R. (2017). Cap-proximal nucleotides via differential eIF4E binding and alternative promoter usage mediate translational response to energy stress. *Elifesciences.Org*. https://doi.org/10.7554/eLife.21907.001

Thiele, A., Nagamine, Y., Hauschildt, S., & Clevers, H. (2006). AU-rich elements and alternative splicing in the β-catenin 3'UTR can influence the human β-catenin mRNA stability. *Experimental Cell Research*, *312*(12), 2367–2378. https://doi.org/10.1016/J.YEXCR.2006.03.029

Tine, M., Kuhl, H., Gagnaire, P. A., Louro, B., Desmarais, E., Martins, R. S. T., Hecht, J., Knaust, F., Belkhir, K., Klages, S., Dieterich, R., Stueber, K., Piferrer, F., Guinand, B., Bierne, N., Volckaert, F. A. M., Bargelloni, L., Power, D. M., Bonhomme, F., … Reinhardt, R. (2014). European sea bass genome and its variation provide insights into adaptation to euryhalinity and speciation. *Nature Communications 2014 5:1*, *5*(1), 1–10. https://doi.org/10.1038/ncomms6770

Tine, M., Kuhl, H., Teske, P. R., & Reinhardt, R. (2021). Genome-wide analysis of European sea bass provides insights into the evolution and functions of single-exon genes. *Ecology and Evolution*, *11*(11), 6546. https://doi.org/10.1002/ECE3.7507

Trapnell, C., Pachter, L., & Salzberg, S. L. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, *25*(9), 1105–1111. https://doi.org/10.1093/BIOINFORMATICS/BTP120

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

Vitting-Seerup, K., Porse, B. T., Sandelin, A., & Waage, J. (2014). SpliceR: An R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. *BMC Bioinformatics*, *15*(1), 1–7. https://doi.org/10.1186/1471-2105-15-81/FIGURES/4

Wilson, G., Aruliah, D. A., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K., Mitchell, I. M., Plumbley, M., Waugh, B., White, E. P., & Wilson, P. (2012). Best Practices for Scientific Computing. *Science*, *340*(6134), 814–815. https://doi.org/10.1371/journal.pbio.1001745

Xia, H., Bi, J., & Li, Y. (2006). Identification of alternative 5′/3′ splice sites based on the mechanism of splice site competition. *Nucleic Acids Research*, *34*(21), 6305–6313. https://doi.org/10.1093/NAR/GKL900

Xing, J., Liu, H., Jiang, W., & Wang, L. (2021). LncRNA-Encoded Peptide: Functions and Predicting Methods. *Frontiers in Oncology*, *10*, 3071. https://doi.org/10.3389/FONC.2020.622294/BIBTEX

Yang, A., Troup, M., & Ho, J. W. K. (2017). Scalability and Validation of Big Data Bioinformatics Software. *Computational and Structural Biotechnology Journal*, *15*, 379–386. https://doi.org/10.1016/J.CSBJ.2017.07.002

Zhang, G., Sun, M., Wang, J., Lei, M., Li, C., Zhao, D., Huang, J., Li, W., Li, S., Li, J., Yang, J., Luo, Y., Hu, S., & Zhang, B. (2019). PacBio full-length cDNA sequencing integrated with RNA-seq reads drastically improves the discovery of splicing transcripts in rice. *The Plant Journal : For Cell and Molecular Biology*, *97*(2), 296–305. https://doi.org/10.1111/TPJ.14120

Zhang, K., Erkan, E. P., Jamalzadeh, S., Dai, J., Andersson, N., Kaipio, K., Lamminen, T., Mansuri, N., Huhtinen, K., Carpén, O., Hietanen, S., Oikkonen, J., Hynninen, J., Virtanen, A., Häkkinen, A., Hautaniemi, S., & Vähärautio, A. (2022). Longitudinal single-cell RNA-seq analysis reveals stress-promoted chemoresistance in metastatic ovarian cancer. *Science Advances*, *8*(8), 1831. https://doi.org/10.1126/SCIADV.ABM1831/SUPPL_FILE/SCIADV.ABM1831_DATA_S1_AND_S2.ZIP

Zhang, Y., Liu, L., Qiu, Q., Zhou, Q., Ding, J., Lu, Y., & Liu, P. (2021). Alternative polyadenylation: methods, mechanism, function, and role in cancer. *Journal of Experimental and Clinical Cancer Research*, *40*(1), 1–19. https://doi.org/10.1186/S13046-021-01852-7/TABLES/3

Zhang, Y., Qian, J., Gu, C., & Yang, Y. (2021). Alternative splicing and cancer: a systematic review. *Signal Transduction and Targeted Therapy 2021 6:1*, *6*(1), 1–14. https://doi.org/10.1038/s41392-021-00486-7

Zhang, Y., Yin, D., & Song, H. (2020). Genome-Wide Identification and Characterization of Gene Families in Arachis: Methods and Strategies. *Frontiers in Genetics*, *11*, 525. https://doi.org/10.3389/FGENE.2020.00525/BIBTEX

Zhu, W., Huang, Y., Zhang, Y., Ding, X., Bai, Y., Liu, Z., & Shen, J. (2022). Identification and characterization of long non-coding RNAs in juvenile and adult skeletal muscle of largemouth bass (Micropterus salmoides). *Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology*, *261*, 110748. https://doi.org/10.1016/J.CBPB.2022.110748