Estudio de mutaciones missense neutrales mediante el uso y análisis de datos ómicos

Albert García López

Máster Universitario de Bioinformática y Bioestadística (Universitat Oberta de Catalunya – Universitat de Barcelona)
Genómica Computacional

Laia Bassaganyas Bars
José Antonio Morán Moreno

5 de junio de 2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Estudio de mutaciones missense neutrales mediante el uso y análisis de datos ómicos*
Nombre del autor:	*Albert García López*
Nombre del consultor/a:	*Laia Bassaganyas Bars*
Nombre del PRA:	*José Antonio Morán Moreno*
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	*Máster Universitario en Bioinformática y Bioestadística (Universitat Oberta de Catalunya – Universitat de Barcelona)*
Área del Trabajo Final:	*Genómica Computacional*
Idioma del trabajo:	*Castellano*
Palabras clave	*Ómicas, mutaciones missense, expresión génica*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Las mutaciones missense neutrales son un tipo de mutación puntual no-sinónima que causan la aparición de un aminoácido distinto, pero de propiedades similares, al péptido que debería haber sido generado por una secuencia de DNA wild-type. Gran parte de las mutaciones asociadas a enfermedades, desde la esclerosis lateral amiotrófica hasta el cáncer, se clasifican como mutaciones missense.

Debido al desconocimiento del impacto de estas mutaciones missense neutrales en enfermedades como el cáncer, este trabajo pretende diseñar un método que permita, en muestras de cáncer colorrectal y utilizando datos provenientes de distintas ómicas, detectar en un grupo de genes seleccionados secuencias peptídicas con mutaciones missense neutrales en los codones que dan lugar a los aminoácidos leucina y arginina.

En este trabajo se ha podido identificar en tres muestras aleatorias de cáncer colorrectal de un dataset de datos proteómicos las secuencias wild-type de tres de los cuatro genes seleccionados mediante el análisis de datos ómicos de diversas fuentes. No obstante, ninguna de las secuencias con la mutación missense introducida se pudieron detectar. Además, se ha conseguido determinar por tests estadísticos que los genes seleccionados con alto contenido GC se expresan en mayor medida que aquellos seleccionados con menor

contenido GC, así como se demostró que no existen diferencias significativas entre los niveles de expresión de los genes problema en células cancerosas y normales.

En general, este trabajo pretende allanar el camino para un método que permita detectar, cuantificar y estudiar mutaciones missense neutrales y determinar su influencia en múltiples procesos patológicos.

Abstract (in English, 250 words or less):

Neutral missense mutations are a class of nonsynonymous point mutations that induce the generation of a different amino acid with similar properties to the one that should have been generated by the wild-type DNA sequence. Most of these disease-associated mutations, from amyotrophic lateral sclerosis to cancer, are classified as missense mutations.

Due to the lack of knowledge regarding the impact of these neutral missense mutations in diseases like cancer, this project has been designed to develop a method to detect, in colorrectal cancer samples and by using data from different omics, neutral missense mutations in codons coding for leucine and arginine peptides in protein sequences generated by a group of selected genes.

In this thesis, by analysing omics data from several sources, wild-type protein sequences from three out of the four genes selected were detected in three random colorrectal cancer samples coming from a dataset of proteomics data. However, none of the sequences containing a neutral missense mutation were identified. Moreover, as it was proved by statistical tests, selected genes with higher amount of GC content were more expressed than those with a lower GC quantity. Additionally, it was also proved by statistical tests that there were no significant differences between the expression levels of selected genes in cancer cells and in normal cells.

Overall, this project aims to pave the way to design a method that allows the detection, quantification and study of neutral missense mutations and understand their influence in multiple pathological processes.

Índice

Agradecimientos

Querría agradecer en primer lugar al Dr. Miquel Àngel Pujana, líder del grupo de investigación basado en el estudio del cáncer de mama, clasificado en el Área de Genética Molecular Humana y Cáncer, dentro del Institut Català d'Oncologia (ICO) por la oportunidad de poder trabajar con él como estudiante de prácticas y por ofrecerme toda la ayuda posible para llevar a cabo este trabajo.

Agradecer también a Luís Palomero, bioinformático integrante del programa ProCURE del Institut Català d'Oncologia (programa en el cual participa también el Dr. Miquel Àngel Pujana) por su asistencia en las tareas más computacionales del trabajo, como la creación de scripts y el procesamiento y análisis de los datos ómicos que en él se trabajan.

Querría también dar las gracias al Dr. Eduard Sabidó, líder de la unidad de Proteómica del CRG (Centre for Genomic Regulation) y la UPF (Universitat Pompeu Fabra), así como a Roger Olivella, miembro del grupo del Dr. Eduard Sabidó, por su disposición en reunirse conmigo y por prestarme toda la ayuda posible para llevar a cabo el análisis proteómico a tiempo para la consecución de este trabajo.

Por último, y no menos importante, querría agradecer el gran apoyo que mi familia me brinda siempre para superar cualquier dificultad a la que me enfrente.

Lista de figuras y tablas

Nota: la mención de cada figura o tabla en este apartado está vinculada con la figura o tabla correspondiente en el trabajo mediante hipervínculos.

para estos genes en muestras tumorales y sanas). Datos extraídos de los archivos `genes_high_gc_cancer_mean.csv` y `genes_high_gc_normal_mean.csv`.

Tabla 5: Lista de los 10 genes con bajo contenido GC seleccionados y con información relevante para su comprensión y comparación: Spectral.Counts de los experimentos PNNL y VU, %GC y valor de expresión de experimentos TCGA para estos genes en muestras tumorales y sanas). Datos extraídos de los archivos `genes_low_gc_cancer_mean.csv` y `genes_low_gc_normal_mean.csv`.

Tabla 6: Lista de los genes seleccionados con alto contenido GC con los valores de expresión génica globales en base a los experimentos realizados por F. Iorio *et al* (2016) [41].

Tabla 7: Lista de los genes seleccionados con bajo contenido GC con los valores de expresión génica globales en base a los experimentos realizados por F. Iorio *et al* (2016) *[41]*.

Tabla 8: Identificación de las secuencias peptídicas wild-type y mutantes asociadas a los genes problema seleccionados (GAPDH, ACTG1, HSP90B1 y FN1) en las muestras del ensayo del Pacific Northwest National Laboratory (PNNL) dentro del estudio del CPTAC Cancer Proteome Confirmatory Colon Study.

Contexto y justificación del Trabajo

Orígenes de la biología de sistemas

Hoy en día, podemos considerar que la biología se ha convertido en una ciencia estrictamente relacionada con el “big data” principalmente debido a los avances en las técnicas high-throughput, las cuales han permitido recolectar, procesar y analizar grandes cantidades de datos biológicos [1], [2]. No obstante, los problemas que ahora surgen no están únicamente relacionados con la ingente cantidad de datos generados sino por la complejidad creciente de estos [1].

A raíz de la necesidad de almacenar, procesar y analizar los datos de diferentes fuentes biológicas nace la biología de sistemas, la cual se fundamenta en el estudio a nivel global o sistémico de una célula o de un organismo mediante métodos experimentales cuantitativos interpretados a través de modelos predictivos matemáticos y estadísticos [1], [3], [4]. A grandes rasgos, la biología de sistemas se puede considerar como una inter-disciplina nacida de la combinación de la biología, la informática, la medicina, la física, la química y la ingeniería [5].

A pesar de que el término parece relativamente novedoso, el concepto de biología de sistemas apareció por primera vez en un artículo de 1999 en relación a la fundación del *Institute for Systems Biology* en Seattle (el cual empezaría su trayectoria en el 2000). No fue hasta este punto cuando la biología de sistemas comenzó a tomar un rol relevante en el escenario actual de las ciencias biológicas (actualmente hay 27.200 resultados en base a la query “Systems Biology” en PubMed) [5].

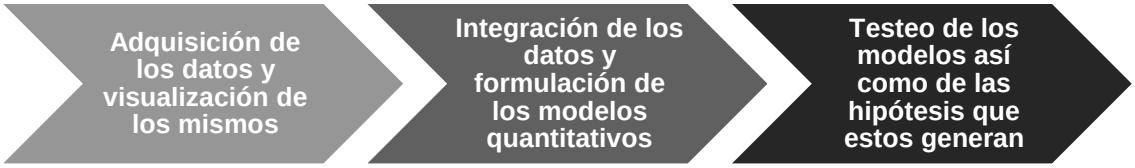
En contraste con la biología clásica, la biología de sistemas no busca únicamente describir los componentes y las propiedades de estos a nivel celular/organismo, sino entender como las múltiples fuentes de información se transmiten y se interpretan por la célula/organismo [1], [3].

Es decir, busca:

1. Entender la estructura de todos los componentes de la célula/organismo hasta el nivel molecular.

2. Predecir el futuro estado de la célula/organismo bajo condiciones normales.
3. Predecir las posibles respuestas dado un determinado estímulo.
4. Estimar los cambios en el comportamiento del sistema a nivel de la perturbación de los componentes del mismo o del ambiente.

Estos objetivos se llevan a cabo a través de una serie de pasos que caracterizan, de forma general, a todo proceso basado en la biología de sistemas [3], [6]–[8]:



Las “ómicas” que integran la biología de sistemas

La biología de sistemas comúnmente se asocia con tecnologías de escala “ómica”, tales como la genómica, la transcriptómica, la proteómica o la metabolómica [3], [9]–[11]. Además, debemos tener en cuenta que la biología de sistemas no sólo recoge esta información “ómica”, sino que además se encarga de procesarla y analizarla [3], [9]–[11].

Podemos definir cada una de estas “ómicas” como (Figura 1):

- **Genómica**: disciplina de la biología molecular que estudia la estructura y función de los genomas. La genómica emplea la combinación de DNA recombinante, métodos de secuenciación del DNA y herramientas bioinformáticas para secuenciar, ensamblar y analizar la estructura y función de los genomas. Por otra parte, la genómica estudia las interacciones entre loci (posiciones físicas fijas en un cromosoma) y alelos (cada una de las formas que puede tomar un gen) [12], [13]. De las numerosas bases de datos existentes, podemos destacar la EMBL del EBI (European Bioinformatics Institute) o la GenBank del NCBI (National Center for Biotechnology Information) [1].
- **Transcriptómica**: disciplina que estudia el transcriptoma (el conjunto total de transcritos de RNA producidos por el genoma mediante el proceso de la transcripción) a través del uso de métodos high-throughput como el

análisis por microarrays o la secuenciación del RNA (RNA-Seq). En este contexto, el mRNA (RNA mensajero) sirve como un intermediario de información, mientras el RNA no codificante lleva a cabo diversas funciones**.** La comparación entre transcriptomas permite la identificación de genes diferencialmente expresados en poblaciones celulares o tejidos distintos (o bien como respuesta a tratamientos diversos) y estudiar cómo estos son regulados, pudiendo además inferir la función de los mismos [14]. Entre las bases de datos más conocidas, podemos destacar Ensembl, nacida de la colaboración entre el EBI y el Wellcome Trust Sanger Institute (Reino Unido), o la Gene Expression Omnibus (GEO) del NCBI [1].

- **Proteómica**: disciplina que analiza el total de proteínas de una célula, tejido u organismo bajo un conjunto de condiciones determinado, con el objetivo de determinar la estructura y el rol de las mismas en el organismo de estudio. Utiliza técnicas y herramientas como las técnicas de fraccionamiento proteico (que permite separar los complejos proteicos o mezclas proteicas), la espectrometría de masas o MS (usada para adquirir los datos necesarios para la identificación de proteínas individuales) y la bioinformática (permite el análisis y el ensamblaje de los datos extraídos de MS) para ofrecer una información detallada del proteoma de un organismo [15], [16]. Existen multitud de bases de datos relacionadas con datos proteicos, aunque a nivel de secuencia destaca UniProt (Universal Protein Resource) y a nivel de estructura destaca PDB (Protein Data Bank) o PROSITE, mientras que a nivel del estudio de interacciones entre proteínas destacan BindingDB y BioGRID [1].
- **Metabolómica**: disciplina que pretende analizar de forma cuantitativa, cualitativa, imparcial y global el metaboloma (el conjunto de metabolitos o pequeñas moléculas de origen químico en una muestra biológica) en un organismo. Dado que los metabolitos pueden ser tanto endógenos (aminoácidos, ácidos orgánicos, ácidos nucleicos, ácidos grasos, vitaminas, carbohidratos…) como exógenos (fármacos, agentes químicos en el ambiente, aditivos alimentarios, toxinas, etc.), la metabolómica engloba un amplio rango de pequeñas moléculas [17]–[20]. En humanos, la base de datos Human Metabolome Database (HMDB) es la encargada

de recolectar la cantidad conocida de metabolitos, la cual contiene la información de 114,100 metabolitos. Por otra parte, bases de datos como KEGG (Kyoto Encyclopedia of Genes and Genomes) permiten dar una visión concisa de las múltiples vías metabólicas conocidas hasta la fecha. A diferencia de ser únicamente participantes en los procesos biológicos, como los genes, transcritos o proteínas, los metabolitos producidos a raíz de estos elementos actúan como vigías de los procesos que ocurren en el organismo de estudio en condiciones fisiológicas o patológicas (por ejemplo, los cambios metabólicos en respuesta a una situación de cáncer). Este hecho sitúa a la metabolómica como la disciplina ómica más relacionada al fenotipo del organismo problema [1], [17].

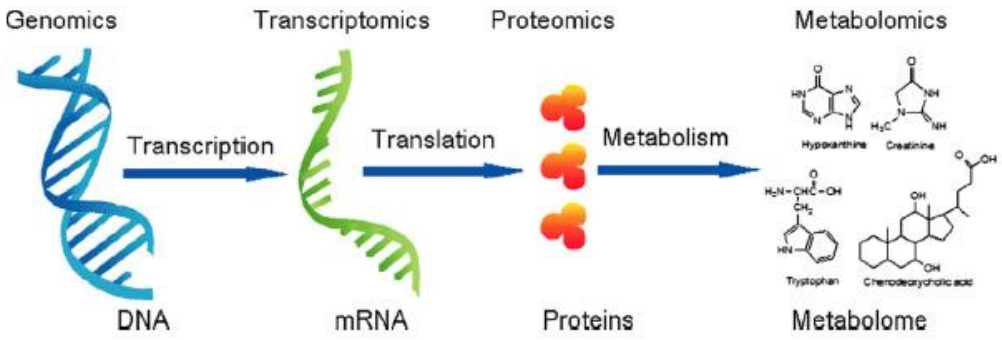


Figura 1 – Representación de las diferentes "ómicas". Como vemos, el flujo de información comienza en la escala genómica (genes) y avanza hasta el nivel metabólico (metaboloma), pasando a través del transcriptoma y el proteoma. (extraído de Zhao, Y.-Y. & Lin, R.-C. 2014).

Tal y como podemos ver en la Figura 2, se muestran los diversos niveles de información biológica (mostrada en la imagen por capas), y cada nivel está compuesto por diversos tipos de moléculas, las cuales dan nombre a las diversas disciplinas "ómicas" mencionadas anteriormente [1]:

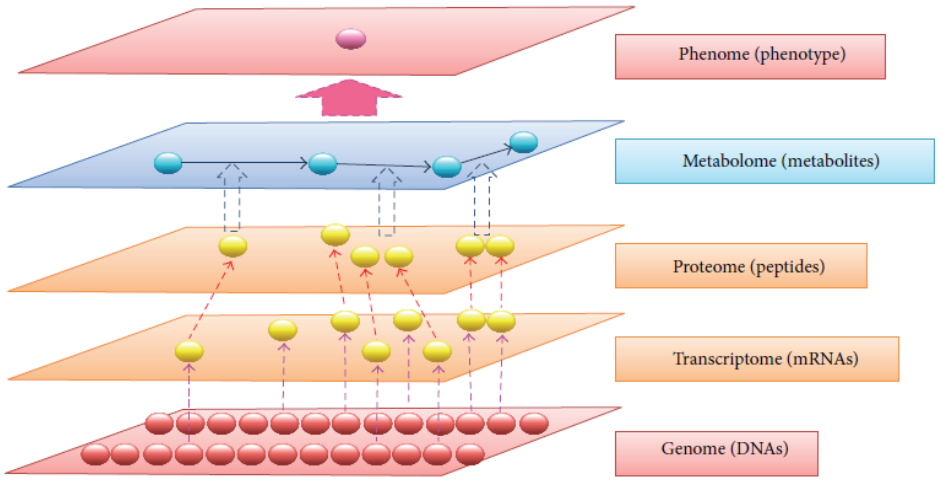


Figura 2 – Tipo de moléculas estudiadas en el campo de la biología de sistemas y su interacción entre ellas (extraído de Altaf-Ul-Amin et al. 2014)

Objetivos y dificultades de la biología de sistemas

El objetivo de la biología de sistemas es, por lo tanto, dar respuesta a la función que llevan a cabo las moléculas de cada capa y como estas interactúan con otras moléculas de la misma capa o de capas diferentes para llevar a cabo funciones biológicas (fenotipo de interés) (ver Figura 2). Por otra parte, la biología de sistemas también permite conocer como la información biológica fluye y se procesa en un sistema biológico dado [1], [3], [5], [7], [8].

A pesar del esfuerzo de la comunidad científica para establecer procedimientos y métodos de identificación de variantes (detectadas y analizadas en genómica y transcriptómica) asociadas a enfermedades, sigue siendo difícil detectar el impacto de estas en ómicas como la proteómica [9], [10].

Esto es debido a que, a diferencia de otras ómicas como la genómica y la transcriptómica, el estudio de la proteómica conlleva un cierto nivel de incertidumbre, rasgo que también comparte la metabolómica [15]. Por ejemplo, si en una célula concreta se expresan 20.000 genes en una situación dada, y suponemos que cada uno de ellos da lugar (de media) a 4 transcritos (mRNA) diferentes, se obtiene un total de 80.000 transcritos. Esta gran variedad de transcritos es resultado de la presencia en los genes de, por ejemplo, variantes de splicing alternativo (splicing del pre-mRNA de diferentes formas, lo cual da

lugar a mRNA de diferentes longitudes y funcionalidades [21]) o SNPs (Single Nucleotide Polymorphisms, polimorfismos causados por mutaciones puntuales que dan lugar a diferentes alelos de un mismo gen que contienen bases diferentes en una posición determinada [22]) [15].

Ahora, si tenemos en cuenta las posibles modificaciones que puede sufrir una proteína a nivel postraduccional (cerca de 300 conocidas, entre ellas las más comunes la fosforilación, glicosilación, metilación o acetilación), imaginemos que se da lugar a una proteína con 4 sitios de fosforilación, la cual da lugar a 16 isoformas de la misma (desde todas sus posiciones fosforiladas a ninguna, 4^2). Si ahora cada uno de los 80.000 transcritos da lugar a una proteína de características similares a la anterior, se daría lugar a 1.280.000 proteínas diferentes (isoformas). A día de hoy, este número supera la capacidad actual de procesamiento y análisis de las tecnologías más avanzadas en escala ómica, lo que conlleva una cierta pérdida de información con cada análisis del proteoma (undersampling) [15].

Análisis del proteoma

Debido a la complejidad intrínseca al tratar con datos proteómicos, existe la necesidad actual de mejorar las metodologías para identificar y caracterizar las variantes proteicas, también llamadas proteoformas [9], [10], dentro de las bases de datos proteómicos actuales. Las modificaciones que se pueden encontrar incluyen variaciones germinales, somáticas y postraduccionales (conocidas también como PTMs) [9], [10], [23]. Para poder explicar la complejidad del proteoma, la espectrometría de masas representa la técnica high-throughput actual a gran escala más relevante y conocida para identificar, caracterizar y cuantificar proteínas y sus variaciones (PTMs, etc.).

La espectrometría de masas (MS) es una técnica analítica que produce espectros (espectro singular) de las masas de las moléculas que componen una muestra. Estos espectros son usados para determinar la composición elemental de la muestra de estudio, así como para conocer la masa de las partículas y de las moléculas que la componen y describir las estructuras químicas de estas, tales como péptidos, metabolitos y otros tipos de compuestos químicos [1], [24]. Por regla general, un espectrómetro de masas consiste en una fuente de iones,

un analizador de masas (que mide la relación masa/carga o m/z) de los analitos ionizados y un detector que registra el número de iones a cada valor m/z. Para volatilizar (pasar péptidos o proteínas a la fase gaseosa sin una degradación significante de la muestra) e ionizar las proteínas, las dos técnicas más usadas en la actualidad son la ionización por electrospray (ESI) y la desorción/ionización láser asistida por matriz (MALDI). Por una parte, ESI ioniza los analitos en una solución y, por ello, está acompañado de técnicas de separación basadas en líquidos (electroforesis en 2 dimensiones en gel de poliacrilamida o 2D PAGE para experimentos tradicionales con proteínas, y cromatografías líquidas o LC para experimentos high-throughput). Estas técnicas de separación permiten distribuir los analitos según su abundancia en la muestra. En general, las técnicas de separación basadas en geles se usan junto con MALDI, mientras que las LC se emplean en conjunto con ESI [1], [23]–[26].

Por otra parte, MALDI sublima (cambio de estado de sólido a gas sin pasar previamente por el estado líquido) e ioniza la muestra en estado de matriz cristalina mediante pulsos laser. A grandes rasgos, MALDI-MS se usa generalmente para analizar mezclas peptídicas simples, mientras que ESI-LC-MS se emplea para el análisis de muestras complejas [1], [23]–[26].

Por lo que respecta al analizador de masas, en el contexto de la proteómica, existen 5 clases principales [24], [27], [28]:

1. **Cuadripolo (Q)**: 4 barras paralelas hiperbólicas en el vacío aplican una radiofrecuencia superpuesta y una corriente eléctrica directa positiva o negativa que provoca que solo aquellos iones con una relación m/z determinada puedan pasar para ser analizados.
2. **Trampa de iones (IT)**: modificación del cuadripolo. En lugar de barras, electrodos generan una corriente eléctrica y una radiofrecuencia donde los iones se capturan/atrapan primero durante un tiempo y luego se les analiza según su relación m/z mediante variaciones en el potencial de la radiofrecuencia.
3. **Time-of-flight o TOF**: consisten en un tubo, donde circulan los iones, y una rejilla de aceleración, que actúa acelerando grupos de iones, que da lugar a un detector. Si dos iones con diferente relación m/z aceleran a

través del tubo libre de fuerzas, el tiempo de llegada al detector será distinto.

4. **Ion-ciclotrón con transformada de Fourier (FT-MS o FT-ICR)**: determinan la relación m/z mediante la frecuencia de ciclotrón de los iones en un campo magnético fijo. Los iones experimentan una fuerza Lorentz que les provoca un movimiento circular perpendicular al campo magnético y gracias a la acción de radiofrecuencias, se incrementa la órbita de ciclotrón y estos pueden ser detectados. Mediante la transformada de Fourier, la intensidad de estos iones detectados se digitaliza respecto al tiempo y se convierten en frecuencias, lo cual permite calcular su relación m/z.
5. **Orbitrap:** similar al FT-ICR, usa la transformada de Fourier para convertir una señal de iones oscilando en frecuencias. No obstante, usan un campo eléctrico para inducir las oscilaciones de los iones en lugar de un campo magnético.

Los parámetros claves de estos analizadores son la sensibilidad, la resolución, la precisión a la hora de cuantificar masas y su capacidad para generar espectros de masa/carga (m/z) o MS/MS, conocido este último también como espectrometría en tándem, en la cual una vez se obtiene el espectro masa/carga de una muestra ionizada por un primer analizador, se selecciona un valor masa/carga destacado del espectro anterior (ion precursor) y se deriva a una célula de colisión, donde un gas inerte (argón, helio u otros gases nobles, dada su baja reactividad) colisiona a nivel de partículas con los iones seleccionados en un proceso denominado “disociación inducida por colisión” [28], [29]. Este proceso genera iones fragmentados (distintos a los iones precursores captados por el primer analizador) que son separados (en relación masa/carga de nuevo) y analizados por un segundo analizador de masas, dando lugar a un espectro de tipo MS (ion precursor)/MS (ion fragmentado) o MS1/MS2, que ofrece una visión más detallada de las proteínas contenidas en la muestra y de su abundancia en la misma [24], [28]–[30].

Existen dos aproximaciones para el análisis cuantitativo de datos proteómicos. En el método más común, llamado “bottom-up” y basado en la proteómica a nivel de péptidos, las muestras son digeridas enzimáticamente (por normal general,

con tripsina) en péptidos antes de su análisis mediante LC-MS. La identificación de los péptidos se lleva a cabo mediante software bioinformático (como SEQUEST, MASCOT o MS-GF) para relacionar los espectros con los patrones de fragmentación teóricos generados usando una base de datos genómica. En el caso del método alternativo, conocido como “top-down”, las proteínas intactas son directamente analizadas mediante LC-MS [23].

Uso de la genómica, la transcriptómica y la proteómica en oncología

Una aplicación fundamental de la espectrometría de masas en el campo de la proteómica consiste en la asociación de datos de espectros proteicos a las secuencias de aminoácidos que los generaron [9]. A raíz de este concepto, la comunidad de investigadores dedicada, por ejemplo, al estudio del cáncer se plantea qué cantidad del genoma de una célula cancerosa, incluyendo sus mutaciones, variaciones estructurales y modificaciones epigenéticas, se representan o tienen un impacto en última instancia en el proteoma de la misma célula [9].

Una clase de estas mutaciones puntuales son las mutaciones missense o con cambio de sentido, las cuales provocan una alteración no-sinónima en un único nucleótido que provoca la aparición de un codón codificante de un aminoácido distinto al que debería ser generado por el codón wild-type. Gran parte de las mutaciones asociadas a enfermedades, entre ellas el cáncer, pertenecen a este grupo de mutaciones missense. A pesar de su ubicua presencia en los genomas, a día de hoy aún no se puede determinar con exactitud si una de estas mutaciones es causante de una enfermedad, dado que tienen que ser estudiadas con cautela a partir de bases de datos de SNPs (polimorfismos de un único nucleótido) detectados mediante análisis estadísticos [31]–[36].

Dado que este tipo de mutaciones cambian el tipo de aminoácido generado, se cree que pueden causar cambios en la función de las proteínas de las que forman parte. Teniendo en cuenta esta premisa, conocer el impacto de este tipo de mutaciones podría ofrecer una mejor comprensión sobre los mecanismos que subyacen bajo enfermedades tan complejas como el cáncer, pudiendo entonces encontrar terapias más efectivas contra estas. No obstante, en ocasiones estas

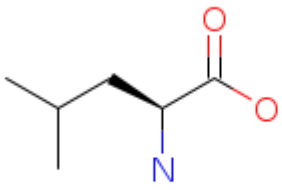
mutaciones parecen pasar desapercibidas en análisis de datos genómicos, transcriptómicos y ciertos estudios proteómicos, dado que en ocasiones pueden no causar cambios relevantes en la proteína codificada, como sería el caso de las mutaciones missense neutrales, que dan lugar a un aminoácido de propiedades químicas y actividad biológica similares al aminoácido generado por la secuencia wild-type [31]–[36].

Con el objetivo de ayudar a determinar si mutaciones missense neutrales pueden tener un impacto en el desarrollo o intensidad de enfermedades como el cáncer, este estudio pretende diseñar un método que permita detectar secuencias peptídicas con mutaciones missense neutrales codificadas por genes expresados en células cancerosas, seleccionados a partir de datos genómicos y transcriptómicos. Esta detección se realizaría usando datos proteómicos generados a partir del análisis de muestras de cáncer colorrectal con espectrometría de masas y mediante la cuantificación y comparación de los picos espectrométricos asociados a la secuencia mutante respecto a los picos espectrométricos producidos por la secuencia wild-type. Pudiendo localizar las regiones donde se encuentran estas mutaciones, en un futuro sería posible realizar estudios dirigidos específicamente contra estas mutaciones y conocer su implicación en múltiples enfermedades humanas, como el cáncer.

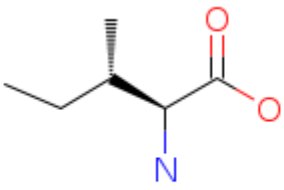
Objetivos del Trabajo

En este trabajo pretendo utilizar los datos genómicos, transcriptómicos y proteómicos obtenidos en bases de datos y en diversos estudios para identificar mutaciones missense neutrales en secuencias peptídicas codificadas por genes expresados en cáncer colorrectal.

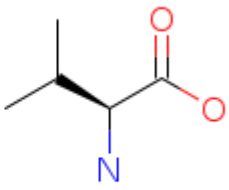
Para este trabajo, se ha centrado la atención en el cambio del aminoácido Leucina (Leu) por Isoleucina (Ile) o Valina (Val) y la mutación del péptido de Arginina (Arg) por Histidina (His) o Lisina (Lys), dado que son cambios entre aminoácidos de estructura química y actividad biológica similares, lo que en el caso de una mutación missense estaría dentro de la categoría de las mutaciones missense neutrales. Por ejemplo, los aminoácidos Leu, Ile y Val son no-polares y contienen residuos alifáticos. Además, sus estructuras químicas son similares [37], [38]:



Leucina (Leu)

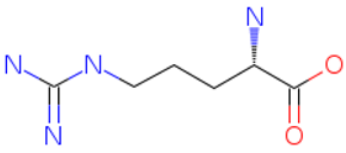


Isoleucina (Ile)

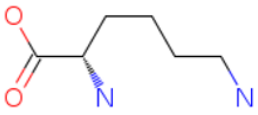


Valina (Val)

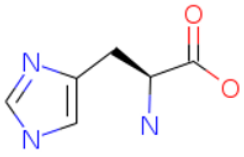
Por otra parte, los residuos Arg, Lys y His están cargados positivamente, lo que les confiere la característica de ser básicos [37], [38]:



Arginina (Arg)



Lisina (Lys)



Histidina (His)

Para poder predecir el cambio de aminoácido, se han seleccionado los codones humanos más infrecuentes para tanto la Leucina (UUA & CUA) como para la Arginina (CGU & CGA), debido que es posible que en condiciones tumorales se favorezca la aparición de estos codones.

Estos codones seleccionados se estudiarán en cuatro genes que se seleccionarán en base a los niveles de expresión génica determinados en datos transcriptómicos de diversas fuentes, el total de espectros identificados para una misma proteína (valor Spectral.Counts en los datos sumario del CPTAC) así como en el contenido GC, ya que es un indicador de la resistencia del gen a la desnaturalización por altas temperaturas (debido a la presencia de enlaces de hidrógeno triples), lo que influye en su funcionalidad. Por otra parte, hay indicios que implican que el porcentaje de GC es un indicativo de los niveles de transcripción del gen y, dependiendo de estos, se podría ver alterada la eficiencia de la traducción de los mRNA del gen en cuestión [39], [40]. Para confirmar esta hipótesis, se seleccionaron 10 genes con un alto contenido GC y 10 genes con un bajo porcentaje de GC y se realizaron una serie de tests estadísticos para comprobar la existencia de diferencias significativas en los niveles de expresión génica entre estos grupos de genes, así como también si estos estaban más o menos expresados en células cancerosas respecto a células normales. A partir de estos grupos de 10 genes seleccionados según su contenido GC, se seleccionarán los dos genes con alto y bajo %GC (total de 4 genes problema) con mayor nivel de expresión génica para realizar, en base a sus correspondientes secuencias peptídicas, el análisis proteómico.

Mediante la base de datos de BioMart (Ensembl), se seleccionará el transcrito de más tamaño (mayor número de pares de bases) y que genere la proteína más grande (mayor número de aminoácidos) de cada gen seleccionado, se descargará la secuencia en formato FASTA y se elegirán, entre las secuencias peptídicas asociadas a los 4 genes problema, 2 residuos de Leucina y 2 de Arginina separados en 10-20 posiciones (upstream y downstream) de otros residuos de Leucina o Arginina, respectivamente. Una vez obtenida estas secuencias wild-type, se copiarán las mismas secuencias y se modificarán los aminoácidos seleccionados por una Val (en el caso de un aminoácido Leu) o por una Lys (en el caso de un residuo Arg), obteniendo así las secuencias mutantes asociadas a los genes problema.

Finalmente, se llevará a cabo el análisis proteómico mediante el software OpenMS y su herramienta TOPPAS (The OpenMS Proteomics Pipeline) usando como referencia los datos proteómicos del estudio “Cancer Proteome

Confirmatory Colon Study" almacenado en el CPTAC (Cancer Institute's Clinical Proteomic Tumor Analysis Consortium).

De esta forma, se pretende poder identificar las secuencias wild-type y mutantes asociadas a los genes problema en los datos proteómicos procedentes de muestras de cáncer colorrectal del CPTAC, realizando así un análisis genómico, transcriptómico y proteómico.

Por lo tanto, se pretende (**Nota:** los orígenes de las diferentes bases de datos y datasets se explican en detalle en la sección de **Materiales**):

- Análisis de datos genómicos (BioMart, Ensembl) y proteómicos (CPTAC Colon Cancer Confirmatory Study) para seleccionar los genes analizados en el estudio del CPTAC en base a su alto o bajo contenido GC y según el total de espectros identificados para una misma proteína (Spectral.Counts).
- Análisis de datos transcriptómicos (datos de expresión génica de F. Iorio *et al* (2016) [41] y RNA-Seq del GDC) para determinar los niveles de expresión génica de los genes seleccionados en células tumorales y en células sanas. Determinar mediante tests estadísticos si las diferencias en los niveles de expresión se deben al contenido GC y/o a la expresión de los genes en células cancerosas o sanas.
- Selección de los transcritos asociados a los genes problema, descarga de su secuencia wild-type en formato .fasta desde la base de datos BioMart de Ensembl, selección de los residuos de Leu y Arg y creación de una secuencia idéntica a la wild-type con la mutación missense neutral introducida.
- Análisis de los datos proteómicos del CPTAC mediante el software OpenMS y TOPPAS e identificación de las secuencias wild-type y mutantes asociadas a los genes problema.

Enfoque y método seguido

A nivel general, los diferentes análisis realizados con los datos genómicos, transcriptómicos y los datos “summary” del estudio proteómico del CPTAC se realizaron mediante scripts diseñados en RStudio, con el lenguaje de programación R. Además, como se verá en detalle en la sección de **Procedimiento y resultados obtenidos**, los tests estadísticos realizados también se llevaron a cabo en RStudio.

Excepto en el caso de la función necesaria para importar datos de naturaleza Rdata y la asistencia en desarrollar una función para extraer y analizar los datos contenidos en BioMart (Ensembl), donde use parte de los scripts redactados previamente por Luís Palomero (IDIBELL), todo el contenido de los scripts en R que permiten la manipulación y análisis de datos procedentes de las distintas ómicas fueron creados por mí.

Se utilizó exclusivamente el lenguaje de programación en R debido a la confianza y habilidad que he ido adquiriendo en el a lo largo del transcurso de este máster.

Por otra parte, la descarga de las secuencias peptídicas de los transcritos asociados a los genes problema se llevó a cabo de forma manual desde la propia página del BioMart (Ensembl).

Por último, el procesamiento y análisis de los datos proteómicos del estudio del CPTAC mediante OpenMS y TOPPAS se realizó a partir de un workflow diseñado por el Dr. Eduard Sabidó (CRG) y Roger Olivella (CRG). Partiendo de su workflow, realicé pequeñas modificaciones que me permitieron procesar y analizar diversos archivos de datos proteómicos de forma simultánea.

Aunque se realizó el workflow con TOPPAS, este es un software para la creación de pipelines algo obsoleto. Actualmente, OpenMS recomienda el uso de KNIME (Konstanz Information Miner), ya que como parte del Center for Integrative Bioinformatics (CiBi), en la German Network for Bioinformatics (deNBI), están centrando sus esfuerzos en la integración de OpenMS en KNIME, el cual contiene todas las herramientas disponibles en TOPPAS. No obstante, en este trabajo se usó TOPPAS ya que el grupo del Dr. Eduard Sabidó tiene la mayoría de sus workflows diseñados en este software, por lo que no tienen aún la misma experiencia en realizar los mismos pipelines con KNIME.

Planificación del Trabajo

Para la consecución de este trabajo, a nivel de recursos, tal y como mencionaba en el apartado anterior, he necesitado los siguientes elementos (estos elementos se detallan con más atención en el apartado **Materiales**):

- **Muestras:** muestras de cáncer colorrectal procesadas en datos proteómicos por el estudio del CPTAC Colon Cancer Confirmatory Study. Estos datos se usaron para la determinación de las mutaciones missense neutrales de Leu y Arg en proteínas codificadas por los genes problema en un contexto de células tumorales.
- **Datos de líneas tumorales:** se usaron los datos procedentes del estudio de F. Iorio *et al* (2016) [41] para conocer la nomenclatura del TCGA usada para clasificar los tipos de cáncer analizados en el estudio.
- **Datos de expresión génica:** del mismo estudio de F. Iorio *et al* (2016) [41] se usaron los datos de expresión génica de los genes analizados en las muestras tumorales tratadas para determinar los niveles de expresión de los genes seleccionados. Los datos no fueron tratados ya que venían normalizados por F. Iorio *et al* (2016) [41].
- **Datos RNA-Seq:** datos transcriptómicos obtenidos en estudios realizados en células cancerosas y sanas almacenados en la base de datos Genomic Data Commons (GDC) del TCGA (The Cancer Genome Atlas). Los datos habían sido previamente normalizados y tratados por el grupo del Dr. Miquel Àngel Pujana.
- **Workflow y base de datos proteómica humana:** el diseño del workflow, así como la base de datos de secuencias proteicas humanas en formato FASTA, fueron creadas por el Dr. Eduard Sabidó y Roger Olivella. Ambos recursos fueron usados en el software TOPPAS 2.0 para identificar las secuencias wild-type y mutantes de los genes seleccionados.

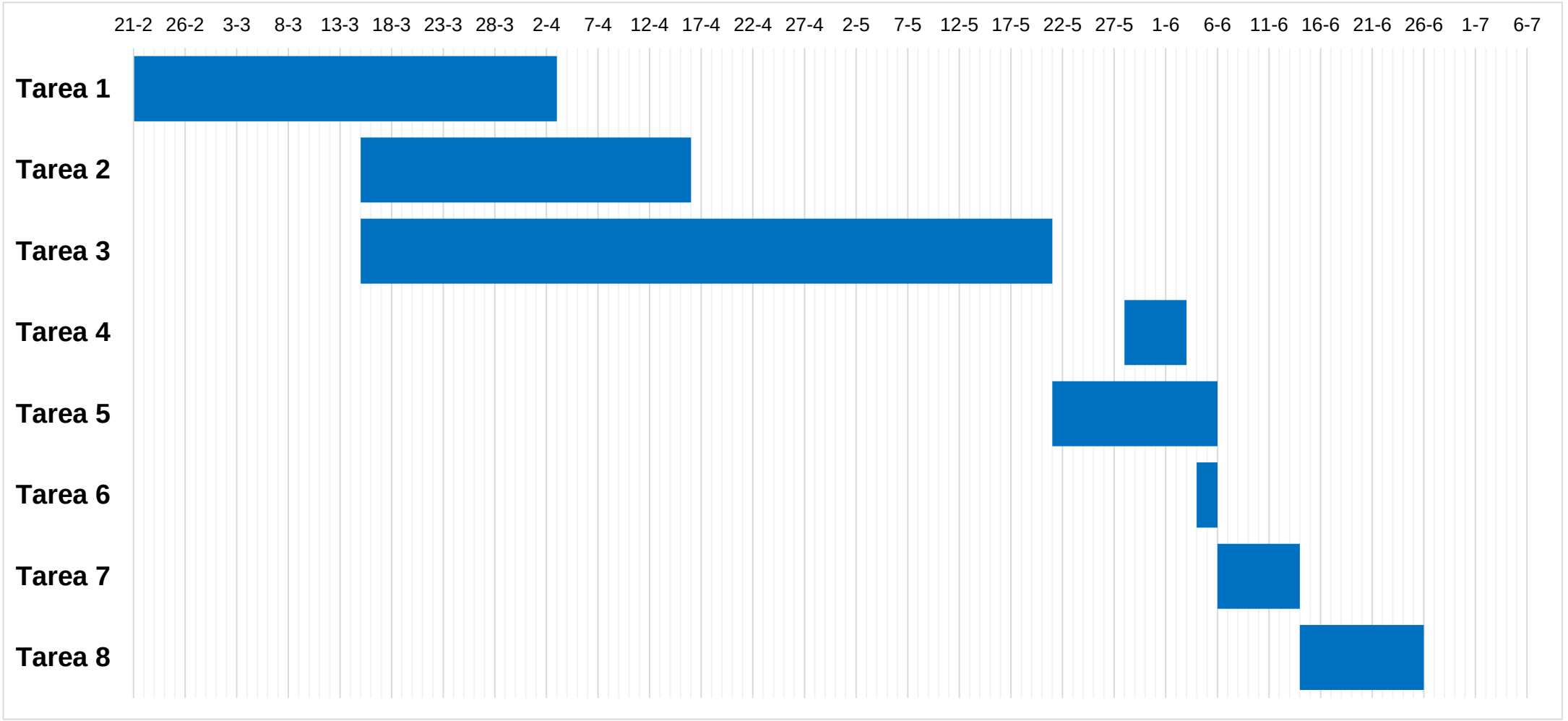
A nivel de metodología, únicamente he necesitado el programa RStudio, un IDE para el lenguaje de programación R, para realizar el procesamiento y análisis de los datos genómicos y transcriptómicos. Para realizar el análisis proteómico, únicamente he necesitado OpenMS 2.0 (que incluye TOPPAS 2.0). En todos los casos, los softwares son de código abierto y gratuitos.

Tareas y planificación temporal

Este trabajo fue organizado inicialmente en 6 tareas, aunque con el avance y las modificaciones del mismo se han ampliado hasta 8. A continuación, añado una tabla con la información específica de cada tarea, la fecha de inicio, la fecha final y la duración total de cada tarea. Finalmente, represento estas tareas en un diagrama de Gantt (página siguiente):

Tareas		Fecha inicio	Fecha final	Duración (días)
Tarea 1	Recogida de información sobre la temática de estudio	21-2	2-4	41
Tarea 2	Aplicación de la metodología para llevar a cabo los análisis genómicos y transcriptómicos	15-3	15-4	32
Tarea 3	Ampliar mis conocimientos en biología de sistemas (ómicas) para el correcto análisis de los resultados obtenidos	15-3	20-5	67
Tarea 4	Aplicación de la metodología para llevar a cabo los análisis proteómicos	28-5	2-6	6
Tarea 5	Redacción de la memoria escrita del Trabajo de Final de Máster	21-5	5-6	16
Tarea 6	Revisión de la memoria escrita	4-6	5-6	2
Tarea 7	Creación y revisión de la presentación oral del trabajo	6-6	13-6	8
Tarea 8	Defensa pública	14-6	25-6	12

Diagrama de Gantt



Breve sumario de productos obtenidos

Durante la realización de este trabajo se han diseñado un total de cuatro scripts en lenguaje de programación R, los cuales pueden encontrarse completos en el apartado de **Anexos** (**Nota**: cada uno de los puntos está enlazado mediante hipervínculos a los correspondientes scripts completos en el apartado **Anexos**):

1. Script para la creación de directorios de trabajo y para las carpetas “data” y “results” donde se almacenaron, respectivamente, los datos y los resultados de los análisis realizados.
2. Script para la lectura de los datos “summary” de los estudios PNNL y VU del ensayo del CPTAC y selección de los genes analizados en función del parámetro Spectral.Counts (el total de espectros identificados para una misma proteína).
3. Script para la extracción de datos genómicos del BioMart (Ensembl) de los genes seleccionados en el script anterior. Selección de los 10 genes problema con alto contenido GC y los 10 genes problema con bajo porcentaje GC. Análisis de los datos de expresión génica de los genes problema en función del dataset RNA-Seq del GDC para células cancerosas y para células sanas. Tests estadísticos para determinar la normalidad, homocedasticidad y la significancia de los valores de expresión génica de los genes seleccionados.
4. Script para la lectura de los datos de líneas tumorales y datos de expresión génica del estudio de F. Iorio *et al* (2016) [41]. Análisis de los niveles de expresión de los 10 genes problema con alto %GC y los 10 genes con bajo %GC. Tests estadísticos para determinar la normalidad, homocedasticidad y significancia de los valores de expresión génica de los genes problema.

Además, los archivos generados a partir de la aplicación de estos scripts, que pueden encontrarse en la carpeta “results”, también son productos de este trabajo.

Por último, el workflow de OpenMS (diseñado por el Dr. Eduard Sabidó y Roger Olivella), así como los archivos idXML generados a raíz del análisis proteómico

de los datos de 3 muestras del ensayo PNNL del CPTAC también fueron productos de esta tesis de máster.

Breve descripción de los otros capítulos de la memoria

Además de los apartados explicados hasta ahora, esta memoria cuenta con un apartado de **Materiales**, donde se explica en detalle las muestras, los datos y los programas usados para realizar los análisis llevados a cabo a lo largo del trabajo.

Adicionalmente, he añadido el apartado **Procedimiento y resultados obtenidos**, en el cual he comentado paso a paso cada etapa del trabajo experimental de la tesis: desde la creación de los directorios de trabajo hasta el análisis proteómico de las muestras del CPTAC, pasando por el procesamiento y análisis de los datos genómicos (BioMart, Ensembl) y transcriptómicos (dataset RNA-Seq del GDC y datos de expresión génica del estudio de F. Iorio *et al* (2016) [41].

Aunque en el apartado de **Anexos** de este trabajo puede encontrarse el código completo de los scripts realizados durante el trabajo, en el apartado anteriormente mencionado de **Procedimiento y resultados obtenidos** explico paso a paso los códigos usados de cada script según el momento de su utilización.

Por último, he añadido un apartado de **Discusión** donde razono y debato los resultados obtenidos a lo largo del estudio. Adicionalmente, al final del trabajo se podrán encontrar también los apartados de **Conclusiones**, donde hago una valoración general del trabajo en base a los resultados obtenidos y a la consecución de los objetivos planteados, así como un apartado de **Glosario**, donde aparecen explicados los términos específicos y acrónimos mencionados en la memoria, y un apartado de **Bibliografía**, donde se citan las fuentes de referencia que he utilizado para contextualizar los conceptos que se tratan en esta tesis.

Destaco por último el apartado de **Agradecimientos**, que he situado al inicio de esta memoria, para agradecer a todas aquellas personas cuya inestimable ayuda ha hecho posible la realización de mi Trabajo de Final de Máster.

Materiales

Muestras

Para llevar a cabo la determinación de las mutaciones missense neutrales en los aminoácidos Leu y Arg de las secuencias peptídicas codificadas por los genes seleccionados en un contexto de célula tumoral, las muestras de cáncer colorrectal se extrajeron de la base de datos CPTAC (Cancer Institute's Clinical Proteomic Tumor Analysis Consortium). Originado en el 2011, el CPTAC es una iniciativa del National Cancer Institute (NCI) dentro del National Institutes of Health (Departamento de Salud y Servicios Humanos, Estados Unidos) que nace de la necesidad de conocimiento sobre las bases moleculares del cáncer mediante el uso de técnicas y metodologías high-throughput para el análisis de proteomas y genomas [42]–[45].

Partiendo de la base de datos original del TCGA (The Cancer Genome Atlas), un programa del NCI para la secuenciación de tumores y el descubrimiento de alteraciones genómicas causantes del cáncer, los más de 30 grupos colaboradores del CPTAC utilizan métodos de digestión por tripsina, metodología "bottom-up" (muestras digeridas con tripsina en péptidos antes de su análisis con LC-MS) y análisis mediante 2D PAGE LC-MS usando analizadores de masas de tipo Orbitrap™. Como resultado, actualmente existen 28 estudios que han generado 10,9 TB de datos proteómicos [42]–[48].

Entre los múltiples estudios existentes, seleccioné el estudio de la Universidad de Vanderbilt (VUMC, dirigido por el Dr. Daniel C. Liebler) y el Pacific Northwest National Laboratory (PNNL, liderado por el Dr. Richard D. Smith) debido a que son una de las colaboraciones que más cantidad y calidad de datos proteómicos han generado y analizado desde los inicios del CPTAC, como podemos comprobar en la Figura 3, donde se compara el número de muestras (azul) y de archivos generados (rojo) con los espectros generados (barras verde y violeta) y con los péptidos diferentes detectados (turquesa), además de con aquellos péptidos distintos con un valor espectrométrico (Spec.Counts o Spectral.Counts, definido como el total de espectros identificados para una misma proteína [49])(color naranja).

Como vemos, en cada uno de estos valores los estudios en cáncer colorrectal destacan entre los demás estudios [43]. Cabe destacar que en la imagen únicamente se menciona a la Universidad de Vanderbilt como única contribuyente del estudio de este cáncer. Esto posiblemente es debido a que, como se puede comprobar en la propia página del estudio en el CPTAC (https://cptac-data-portal.georgetown.edu/cptac/s/S037), los datos proteómicos de la Universidad de Vanderbilt (VU) van del 2015 al 2016, mientras que los del Pacific Northwest National Laboratory (PNNL) no aparecen hasta 2017. Dado que la imagen adjunta es de 2016, el Pacific Northwest National Laboratory aún no había contribuido en este estudio junto con la Universidad de Vanderbilt, por lo que podríamos esperar una mejora sustancial (actualmente, los datos generados por el estudio llegan a un total de 1,1 TB) en los factores mencionados anteriormente respecto a los de la imagen, con lo cual refuerza la posición de este estudio como uno de los más prolíficos existentes hasta la fecha en el CPTAC [50].

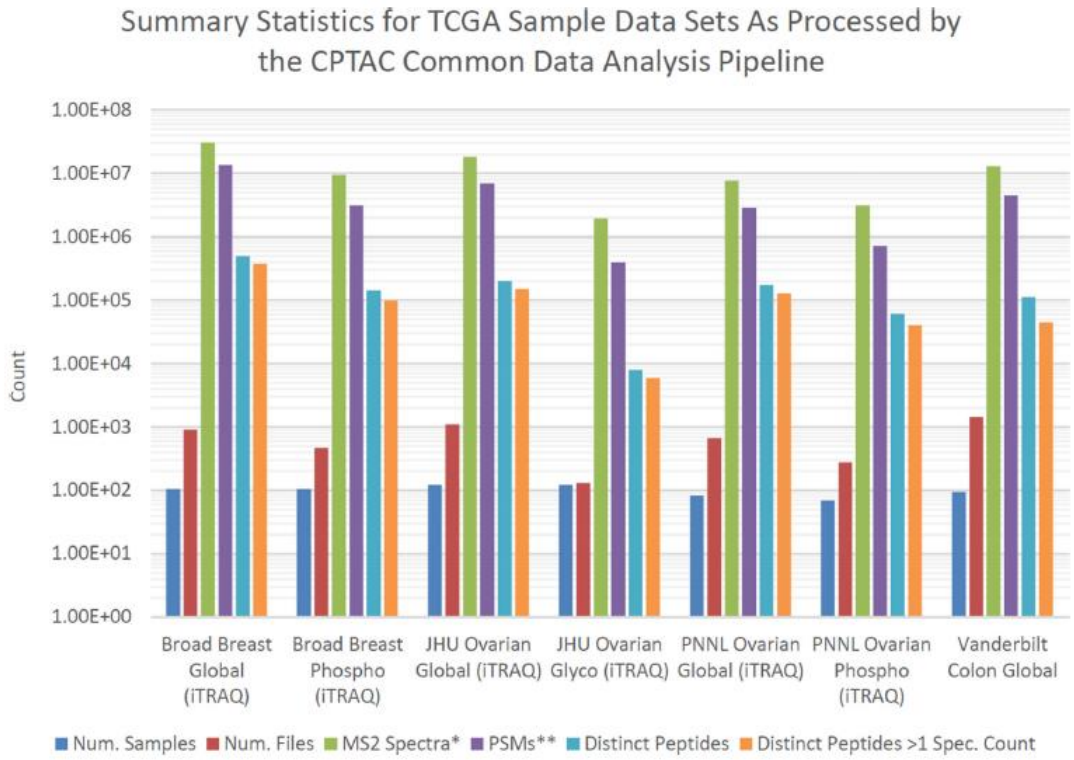


*Figura 3 – Conjunto de los mayores análisis del CPTAC usando muestras pertenecientes a TCGA. *El contaje de espectros de MS2 proviene de los ficheros MGF (Mascot generic format, un tipo de archivo de lectura de datos de espectrometría de masas) del NIST (National Institute for Standards and Technology). **El FDR (False Discovery Rate, o ratio de falsos positivos, el cual es un valor de corrección para comparaciones múltiples en que, si el ratio es de 0,05, indica que el 5% de los positivos son realmente negativos* [51], [52]*) a nivel de PSM (Peptide-Spectrum Match), los contajes del cual excluyen las identificaciones marcadas como ambiguas (por ejemplo, >1 valor de parecido/match de péptidos). Gráfico extraído de P. A. Rudnick et al (2016)* [43].

Tal y como podemos comprobar en la página del estudio (link anterior), 104 muestras de cáncer colorrectal fueron estudiadas por ambos centros. 100 tumores fueron analizados en la Universidad de Vanderbilt mediante la técnica conocida como “label-free global proteomic profiling” (LFQ). Este método emplea el área del pico espectrométrico del ion precursor (MS1) para cuantificar la abundancia del péptido en concreto. Al contrario del marcaje de isotopos con técnicas de fluorescencia (tradicionalmente usadas), en las técnicas “label-free” se prepara cada muestra por separado (en lugar de todas combinadas y marcadas como es el caso del método tradicional por labelling) para ser separadas por LC-MS individuales [53], [54].

Alternativamente, también por la VUMC, 97 tumores se cuantificaron mediante el método de labelling Tandem Mass Tags o TMT (kit Tandem Mass Tag™ o TMT™ 10-plex de ThermoFisher). Este método, junto con iTRAQ (Isobaric Tags for Relative and Absolute Quantification), permiten procesar entre 6 y 8 muestras (respectivamente). En estos casos, los péptidos son químicamente marcados con reactivos isobáricos (que tienen la misma masa) de diversas masas. Las muestras son separadas en conjunto y son sometidas a análisis de MS [53], [55]. Por último, 93 muestras tumorales se analizaron con ambos métodos.

Por otra parte, en el caso del estudio del PNNL, 100 muestras de colon sano fueron evaluados por TMT, con 96 muestras normales emparejadas con muestras tumorales del mismo participante. Las 4 muestras restantes de colon normal del estudio del PNNL se emparejaron con 4 muestras tumorales ensayadas previamente por VUMC. En general, estos datos combinan el estudio proteómico de muestras de tumor y normales de 100 individuos.

Además de los datos proteómicos, se realizó un análisis del fosfoproteoma de las 197 muestras tratadas anteriormente (97 tumorales y 100 sanas). En este caso, se analizó la extensión y la naturaleza de la fosforilación proteica mediada por quinasas [56].

Los datos generados por el estudio, por el momento, engloban archivos de tipo [50], [57]:

- **raw:** archivos en bruto sacados directamente del MS.
- **mzML:** formato estandarizado por HUPO Proteomics Standards, una comunidad centrada en el estudio de la proteómica y la interactómica que define los estándares de los datos para su correcta visualización y circulación [58]. Este tipo de archivo puede ser leído por software como *OpenMS* para analizar espectros proteicos.
- **PSM:** Peptide-Spectrum Match. Datos donde se visualiza el análisis del espectro proteico y las coincidencias entre el espectro MS/MS y las secuencias peptídicas de las proteínas analizadas. Estos datos son tratados por los Proteome Characterization Centers (PCCs), incluidos dentro del CPTAC [59].
- **prot:** datos referentes al ensamblado proteico y la abundancia relativa de las proteínas de las muestras analizadas.
- **meta:** datos clínicos de pacientes, mapeado de las muestras con los marcajes por iTRAQ o TMT, etc.

Para llevar a cabo el análisis de las secuencias de aminoácidos de los genes seleccionados expresados en cáncer colorrectal e identificar la presencia de mutaciones missense neutrales, usaremos los datos **mzML**. Para ello, nos serviremos del software **OpenMS** y **TOPPAS.**

OpenMS

OpenMS es un software de código abierto, basado en C++ y Python, para el tratamiento y análisis de datos LC-MS, la cual cosa ayuda a garantizar una reproducibilidad en los estudios de datos high-throughput proteómicos, así como también metabolómicos [60]. Para agilizar e intensificar el proceso de análisis, OpenMS se ha integrado con TOPPAS, un software que permite la creación de workflows para el análisis de datos biológicos. A partir de estos dos programas, se crea un workflow (TOPPAS) que permite la integración de los archivos mzML y el análisis de estos mediante OpenMS.

Datos de líneas tumorales (F. Iorio)

Para conseguir la información de las líneas celulares de cáncer y seleccionar los genes que se expresen en células de cáncer colorrectal, se utilizaron los

resultados del estudio de F. Iorio *et al* (2016) [41], donde se identifican alteraciones derivadas por el cáncer en 11.289 tumores procedentes de 29 tejidos. En estas alteraciones se tienen en cuenta mutaciones somáticas, variaciones en el número de copias (CNV), metilaciones del DNA y cambios en la expresión génica. Las alteraciones encontradas en estas muestras pudieron ser mapeadas a 1.001 líneas celulares de cáncer humanas y correlacionadas con 256 fármacos.

Dado que nuestro objetivo es tratar de identificar mutaciones missense neutrales en células de cáncer colorrectal, usamos la información referente a estas líneas celulares tumorales para llevar a cabo nuestro estudio. La información de las líneas celulares tumorales es la siguiente (Tabla 1):

TCGA Label	Definition
ACC	Adrenocortical carcinoma
ALL	Acute lymphoblastic leukemia
BLCA	Bladder Urothelial Carcinoma
BRCA	Breast invasive carcinoma
CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma
CLL	Chronic Lymphocytic Leukemia
COREAD	Colon adenocarcinoma and Rectum adenocarcinoma
DLBC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma
ESCA	Esophageal carcinoma
GBM	Glioblastoma multiforme
HNSC	Head and Neck squamous cell carcinoma
KIRC	Kidney renal clear cell carcinoma
LAML	Acute Myeloid Leukemia
LCML	Chronic Myelogenous Leukemia
LGG	Brain Lower Grade Glioma
LIHC	Liver hepatocellular carcinoma
LUAD	Lung adenocarcinoma
LUSC	Lung squamous cell carcinoma
MB	Medulloblastoma
MESO	Mesothelioma
MM	Multiple Myeloma
NB	Neuroblastoma
OV	Ovarian serous cystadenocarcinoma
PAAD	Pancreatic adenocarcinoma
PRAD	Prostate adenocarcinoma
SCLC	Small Cell Lung Cancer

SKCM	Skin Cutaneous Melanoma
STAD	Stomach adenocarcinoma
THCA	Thyroid carcinoma
UCEC	Uterine Corpus Endometrial Carcinoma

Tabla 1 – Lista de los tipos de cáncer estudiados en el experimento de F. Iorio et al (2016) [41]*, relacionada con la Figura 1 del mismo artículo* [41]*. Lista extraída y levemente modificada del trabajo de F. Iorio et al (2016)* [41].

Datos de expresión génica (F. Iorio)

De la misma forma que con los datos de las líneas tumorales, se utilizaron los datos de la expresión génica de los genes estudiados en el trabajo de F. Iorio *et al* (2016) [41] para llevar a cabo este estudio. Estos datos se descargaron desde la página del grupo de F. Iorio *et al* en la base de datos Genomics of Drug Sensitivity in Cancer (fundada por el Wellcome Sanger Institute, UK y el Massachusetts General Hospital Cancer Center, USA) en el enlace siguiente (https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources//Data/preprocessed/Cell_line_RMA_proc_basalExp.txt.zip). En este caso, los datos no fueron transformados de ninguna forma adicional ya que venían normalizados por el propio grupo de F. Iorio *et al* [41].

Datos RNA-Seq del GDC

Una vez seleccionados los genes problema en base a los datos del CPTAC y BioMart (Ensembl), se usaron los datos de RNA-Seq de células tumorales y sanas para estudiar la posible expresión diferencial de los genes problema en un contexto de cáncer y en una situación de condiciones normales (células sanas). En este caso, los datos RNA-Seq provienen de experimentos con células sanas y cancerosas (de tipo cáncer de mama, asociado al gen BRCA) almacenados en la base de datos GDC (Genomic Data Commons), que pertenece al TCGA (The Cancer Genome Atlas). Estos datos fueron normalizados previamente por miembros del grupo del Dr. Miquel Àngel Pujana mediante el método FPKM-UQ (Fragments Per Kilobase of transcript per Million mapped reads upper quartile), implementado en el GDC a nivel de lecturas genómicas producidas por el método HTSeq (secuenciación high-throughput), una librería de código abierto en lenguaje Python que permite el tratamiento y manipulación de datos genómicos

y RNA-Seq [61]. La fórmula usada para la generación de estos valores FPKM-UQ se basa en:

$$\mathbf{FPKM = [RM_g * 10^9] / [RM_{75} * L]}$$

Donde $\mathbf{RM_g}$ es el número de lecturas mapeadas a un gen, $\mathbf{RM_{75}}$ es el número de lecturas mapeadas en los genes localizados en el 75º percentil, y **L** es la longitud de los genes en pares de bases. El $\mathbf{10^9}$ se emplea para normalizar los valores a "kilo bases" y a "millones de lecturas mapeadas" [62]. Por último, en el caso particular del grupo del Dr. Miquel Àngel Pujana, se realizó un paso adicional donde los valores se transformaron mediante un **log2**, para mejorar su comprensión.

Procedimiento y resultados obtenidos

Creación de los directorios de trabajo

Este paso inicial, aunque sencillo, se realizó para facilitar el proceso de almacenamiento de datos y de resultados. Para ello, al igual que para el resto de análisis realizados, se utilizó el lenguaje de programación R y su interfaz más comúnmente utilizada, RStudio. R es un lenguaje de programación ampliamente usado para llevar a cabo análisis de estadística computacional y representación gráfica de datos estadísticos. Debido a su celebridad en el campo de la bioinformática y la bioestadística, y los conocimientos que he ido adquiriendo en su uso a lo largo del máster, decidí realizar los análisis que han llevado a la consecución de este trabajo en este lenguaje. Por otra parte, se usó RStudio, una IDE (Integrated Development Environment) para el lenguaje R que incluye una consola, un editor para la redacción de scripts (soporta además la ejecución directa de código) así como herramientas de visualización de gráficos, historial, manejo del espacio de trabajo, etc. Tanto R como RStudio son softwares de código abierto y gratuitos [63], [64].

Como paso inicial, se designó una carpeta de mi ordenador personal como directorio de trabajo y se le asignó una variable dentro de R (`workingDir`):

```
setwd("D:/Universidad/Máster Oficial Bioinformática y Bioestadística
UOC-UB/2o Semestre/Treball de Final de Máster/Scripts")

workingDir <- getwd()
```

De forma similar al paso anterior, se generan los directorios `data` y `results` para almacenar, respectivamente, los datos y los resultados del trabajo. Finalmente, creamos dos variables independientes para dirigir los resultados a los directorios creados y denominarlas en base a datos o resultados (`dataDir` y `resultsDir`):

```
system("mkdir data")

system("mkdir results")

dataDir <- file.path(workingDir, "data")

resultsDir <- file.path(workingDir, "results")
```

Selección de los genes problema

Una vez creados el directorio de trabajo y los directorios para los datos y los resultados, seleccionamos los genes problema para estudiar la presencia de mutaciones missense neutrales en los aminoácidos Leu y Arg. Para ello, importamos los archivos “summary” de los estudios PNNL y VU del estudio del CPTAC. Usamos el siguiente código:

```
data_pnnl_summary <- read.table(file = 'D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/data/CPTAC COprospective PNNL Proteome CDAP Protein Rep
ort_r1/CPTAC2_Colon_Prospective_Collection_PNNL_Proteome_summary.tsv',
sep = '\t', header = TRUE)

data_vu_summary <- read.table(file = 'D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/data/CPTAC_COprospective_VU_Proteome_CDAP_Protein_Repor
t_r1/CPTAC2_Prospective_Colon_VU_Proteome_summary.tsv', sep = '\t',
header = TRUE)
```

En este caso utilicé la dirección concreta de cada uno de los archivos en lugar de recurrir a la carpeta de datos creada en el punto anterior debido a que RStudio me lanzaba un error que posiblemente se debe a la instalación del software en mi ordenador.

Una vez importados los datos “summary”, podemos comprobar que información contienen con el siguiente código:

```
str(data_pnnl_summary)

str(data_vu_summary)
```

De esta forma, comprobamos que la primera columna en ambos casos coincide con los genes estudiados, seguida por los valores Spectral.Counts (el total de espectros identificados para una misma proteína), Distinct.Peptides (péptidos únicos encontrados en el análisis MS) y Unshared.Peptides (péptidos no compartidos detectados mediante el análisis MS) para cada muestra analizada. En las últimas columnas encontramos los valores totales de Spectral.Count, Distinct.Peptides y Unshared.Peptides, así como la referencia del gen en la base de datos del NCBI (NCBIGeneID), su código “Authority”, la descripción del producto del gen, el organismo, el cromosoma, el locus, el código proteico y los ensayos realizados.

Dado que únicamente nos interesa conseguir la información de los genes y los resultados globales de los análisis MS, emplearemos la librería `dplyr` (del paquete `tidyverse`) para filtrar los datasets según nuestras necesidades. Esta librería (`dplyr`) permite la manipulación de datos gramáticos, es decir, no modifica datos numéricos para hacer el filtraje [65]. Aplicamos el siguiente código para obligar a RStudio a usar `dplyr`:

```
if(!(require(dplyr))) install.packages("tidyverse")

library(dplyr)
```

A continuación, hacemos el filtraje de las columnas deseadas tanto para los datos de PNNL como para los de VU (usando la función `select` de `dplyr`):

```
data_pnnl <- select(data_pnnl_summary,
                    Gene,
                    Spectral.Counts,
                    Distinct.Peptides,
                    Unshared.Peptides,
                    NCBIGeneID,
                    Authority,
                    Description,
                    Organism,
                    Chromosome,
                    Locus,
                    Proteins,
                    Assays)

data_vu <- select(data_vu_summary,
                    Gene,
                    Spectral.Counts,
                    Distinct.Peptides,
                    Unshared.Peptides,
                    NCBIGeneID,
                    Authority,
```

```
                    Description,
                    Organism,
                    Chromosome,
                    Locus,
                    Proteins,
                    Assays)
```

Con este filtraje, se obtienen los 4685 genes para PNNL y los 4622 genes para VU con las columnas deseadas. No obstante, debemos aplicar un mínimo de Spectral.Counts para que los espectros de las proteínas generadas por los genes sean visibles a la hora de realizar el análisis proteómico de las mismas. Por ello, filtramos aquellos genes que tengan un valor total de Spectral.Counts superior o igual a 100. Aplicamos el siguiente código para llevar a cabo esta acción:

```
data_pnnl_sel <- filter(data_pnnl, Spectral.Counts >= 100)

data_vu_sel <- filter(data_vu, Spectral.Counts >= 100)
```

En estas nuevas variables, obtenemos 2805 genes para PNNL y 2824 genes para VU que cumplen esta condición.

Seguidamente, combinaremos ambos datasets (función `merge` de `dplyr`) en uno mismo y filtraremos por los genes que estén en ambos datasets (columna “Gene” en los dos datasets). Para poder diferenciar entre los parámetros de cada gen encontrados en los dos estudios de forma independiente, añadiremos a estas columnas los sufijos “.PNNL” y “.VU”:

```
total_data <- merge(data_pnnl_sel, data_vu_sel, by = "Gene", suffixes
= c(".PNNL", ".VU"))
```

Una vez hecha la selección de los genes que se encuentran en ambos datasets, obtenemos 2198 genes en común.

Seguidamente, esta variable es exportada como un archivo .csv (valores separados por comas) con el nombre `listagenestotal.csv` mediante el comando:

```
write.csv(total_data, "D:/Universidad//Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenestotal.csv")
```

Por último, designamos una nueva variable llamada `total_genes`, donde se almacenará (en forma de vector) los nombres de los genes filtrados en ambos datasets y, como en el paso anterior, exportamos esta lista en formato csv en un archivo llamado `listagenes.csv`:

```
total_genes <- as.vector(total_data$Gene)
write.csv(total_genes, "D:/Universidad//Máster Oficial Bioinformática
y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenes.csv")
```

Extracción de datos de BioMart

En este apartado, debemos extraer la información genómica de los genes seleccionados en el apartado anterior (analizados en el estudio del CPTAC) de la base de datos de BioMart, dentro de Ensembl. La base de datos Ensembl fue creada en 1999 con el objetivo de anotar genomas, integrar estas anotaciones con otros datos biológicos y difundir públicamente estos datos en su página web (https://www.ensembl.org). De los diferentes grupos que componen Ensembl, el equipo Genebuild se dedica al tratamiento, almacenamiento y análisis de los genomas de diversas especies, así como de mantener BioMart, un sistema de código abierto de bases de datos que unifica los datos de muchas bases de datos (40 aproximadamente, entre ellas Ensembl Genomes, la base de datos de cáncer COSMIC, HapMap, UniProt, etc.) para facilitar el acceso a la información allí contenida [66], [67].

Para llevar a cabo el objetivo comentado anteriormente, creamos un script donde en primer lugar se activen las librerías `dplyr` y `data.table`, necesarias para llevar a cabo el procedimiento. Además, mediante Bioconductor (función `biocLite`), instalamos el paquete de BioMart ("`biomaRt`"). Usamos el siguiente código:

```
if(!require(biomaRt, dplyr, data.table)){
  source("http://bioconductor.org/biocLite.R")
  biocLite()
  biocLite("biomaRt")
  require(biomaRt)
  require(dplyr)
  require(data.table)
}
```

A continuación, mediante la modificación de un script sencillo creado por el grupo del Dr. Miquel Àngel Pujana, creamos una función llamada `mapHgncSymbols` donde se extrae (`mapping`) los parámetros `"entrezgene", "hgnc_symbol","ensembl_gene_id","gene_biotype","percentage_gene_gc_content",` se filtra por `"hgnc_symbol"` y como resultado se da el valor `symbol`, que viene a ser el nombre del gen asociado a estos parámetros:

```
mapHgncSymbols <- function(ensembl, symbols){
  mapping <- getBM(attributes = c("entrezgene", "hgnc_symbol",
"ensembl_gene_id","gene_biotype","percentage_gene_gc_content"),
                   filters = "hgnc_symbol" , values = symbols,
                   mart = ensembl)
  return(mapping)
}
```

A continuación, creamos la variable `ensembl` que, primero recoja el parámetro ensembl mediante la función `useMart` y, a continuación, seleccione el dataset (`useDataset`) de genes humanos de Ensembl.

```
ensembl = useMart('ensembl')
ensembl = useDataset("hsapiens_gene_ensembl", mart=ensembl)
```

A continuación, para obtener un data frame con los nombres de los genes seleccionados en el apartado anterior, creamos una variable llamada `symbols` (nombres de los genes) que importe el archivo `listagenes.csv` y cambie el nombre de la columna con los nombres de los genes por `SYMBOL`.

```
symbols = as.data.frame(
  read.csv(
    "D:/Universidad/Máster Oficial Bioinformática y Bioestadística
UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenes.csv",
    row.names = 1,
    col.names = c("", "SYMBOL"),
    stringsAsFactors = FALSE
    )
  )
```

Una vez hecho este paso, usamos la función creada previamente `mapHgncSymbols` para utilizar los datos genómicos de los genes de Ensembl (`ensembl`) y cruzarlos con los nombres de los genes problema del CPTAC (`symbols$SYMBOL`). Almacenamos esta información en la variable `genes` y la exportamos como archivo .csv con el nombre `genesGC`. Les llamamos así

porque incluyen información relevante para el estudio como el contenido GC (el porcentaje de pares de bases con guanina o citosina), que permite dar una indicación de la resistencia del gen a la desnaturalización por temperaturas elevadas (debido a la más alta cantidad de enlaces de hidrogeno triples), la cual cosa afecta a la funcionalidad del mismo. Además, el porcentaje de GC es un indicativo de los niveles de transcripción del gen y, dependiendo de estos, se podría ver alterada la eficiencia de la traducción de los mRNA del gen en cuestión [39], [40].

```
genes = mapHgncSymbols(ensembl, symbols$SYMBOL)
data_genes = as.data.frame(genes)
write.csv(data_genes, "D:/Universidad/Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genesGC.csv")
```

Selección de genes de alto y bajo contenido GC

Seguidamente, mezclamos con `dplyr` los archivos `listagenestotal.csv` (variable `total_data_final`) con la variable `genes_gc` (que incorpora los genes seleccionados con los datos genómicos de BioMart). No obstante, para poder realizar el filtraje, debemos cambiar el nombre de la columna de `genes_gc` por "Gene" (misma columna que tiene la variable `total_data_final`):

```
total_data_final <- read.csv("D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenestotal.csv", row.names = 1)

genes_gc <- read.csv("D:/Universidad/Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genesGC.csv",row.names = 1)

colnames(genes_gc)[2] <- "Gene"
genes_gc_score <- merge(total_data_final, genes_gc, by = "Gene") #Aquí
tendríamos combinado todo, GC y datos espectrales.
```

Relacionando ambos datasets (variable `genes_gc_score`), obtenemos la información para 2469 genes, entre los cuales tenemos los 2198 genes seleccionados mediante los datos del estudio del CPTAC, ahora con información sobre los datos MS y sobre datos genómicos de Ensembl. Por último,

organizamos los 2469 genes según su contenido GC en orden ascendente (función `arrange`):

```
genes_gc_score <- arrange(genes_gc_score,
genes_gc_score$percentage_gene_gc_content)
```

A continuación, mediante el uso de `dplyr`, seleccionaremos 10 genes con alto contenido GC y otros 10 genes con bajo contenido GC. Esto es debido a que, como explicaba anteriormente en relación al contenido GC, estos genes podrían tener diferentes niveles de transcripción, lo cual podrá alterar la eficiencia de la traducción de los mRNA que den lugar. Para ello, filtraremos los genes que tengan un valor Spectral.Counts superior a 7500 para genes con alto %GC y superior a 6600 para los genes con bajo contenido GC. Además, los genes con alto %GC deberán tener un contenido en GC igual o superior al 60%, mientras que los genes con bajo %GC deberán tener un nivel inferior o igual al 40%. Los resultados de este filtraje se almacenarán y se exportarán (formato .csv) en ficheros separados (`high_gc_genes.csv` y `low_gc_genes.csv`)

```
highgc <- filter(genes_gc_score,
                 genes_gc_score$Spectral.Counts.PNNL > 7500,
                 genes_gc_score$percentage_gene_gc_content >= 60
                 )

write.csv(highgc, "D:/Universidad/Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/high_gc_genes.csv")

lowgc <- filter(genes_gc_score,
                genes_gc_score$Spectral.Counts.PNNL > 6600,
                genes_gc_score$percentage_gene_gc_content <= 40
                )

write.csv(lowgc, "D:/Universidad/Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/low_gc_genes.csv")
```

Los 10 genes con alto contenido GC (ordenados de menor a mayor) seleccionados son (columnas extraídas de `high_gc_genes.csv`) (Tabla 2):

	Gene	Spectral.Counts.PNNL	Spectral.Counts.VU	percentage_gene_gc_content
1	GAPDH	13854	9545	60.61
2	FLNC	12860	9737	60.86
3	ACTG1	53597	22759	61.09
4	HIST2H2AA3	7978	5269	61.75
5	FLNA	43223	35387	62.28
6	HIST1H3D	13036	9197	62.29
7	ACTA1	59482	19259	62.54
8	COL6A1	7843	7802	64.67
9	H2AFX	8192	5827	66.42
10	HBA1	23121	10219	67.26

Tabla 2 – Lista de los 10 genes con alto contenido GC seleccionados.

Por otra parte, los 10 genes con bajo contenido GC (ordenados de menor a mayor) seleccionados son (columnas extraídas de `low_gc_genes.csv`) (Tabla 3):

	Gene	Spectral.Counts.PNNL	Spectral.Counts.VU	percentage_gene_gc_content
1	ALB	66797	26718	34.9
2	FGB	7603	7476	35.09
3	COL12A1	6625	11461	36.87
4	A2M	6885	6568	37.18
5	COL1A2	6782	5142	37.57
6	HBB	34585	21394	37.64
7	HBD	17234	15031	38.32
8	FBN1	12992	9741	38.92
9	FN1	14726	17853	39.44
10	HSP90B1	7882	8569	39.78

Tabla 3 - Lista de los 10 genes con bajo contenido GC seleccionados.

Uso de datos RNA-Seq del GDC y filtraje de los genes problema

En este punto, usaremos los datos RNA-Seq de células tumorales y sanas de la base de datos GDC, normalizados y ajustados previamente por el grupo del Dr. Miquel Àngel Pujana, para estudiar los niveles de expresión de los genes problema, seleccionados en el apartado anterior según su contenido GC. En primer lugar, usamos una función creada originalmente también por el grupo del Dr. Miquel Àngel Pujana para importar los datos RNA-Seq a RStudio. Esto es debido a que RStudio detecta con el mismo nombre los dos archivos RNA-Seq (para células tumorales y células sanas) e impide importar ambos archivos a la

vez. La función trabaja sobre una dirección (`path`) en base a una variable, cuyo uso permite la carga del archivo de interés. Por último, la función nos consigue la variable, consiguiendo a la vez importar el archivo que llama esta:

```
loadRData = function(path){
  variable = load(path)
  return(get(variable))
}
```

Creada una vez la función, importamos los datos RNA-Seq y los asociamos a variables independientes:

```
rnaseqcancer <- loadRData("D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/rnaSeq.RData")

rnaseqnormal <- loadRData("D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/rnaSeqNormals.RData")
```

A continuación, mezclamos (función `merge` de `dplyr`) los datos de los datasets de genes con alto contenido GC y de bajo contenido GC con los datasets de RNA-Seq para células normales y para células tumorales en función de la columna común entre datasets "`ensembl_gene_id`".

```
genes_high_gc_cancer <- merge(highgc, rnaseqcancer, by =
"ensembl_gene_id")
genes_high_gc_normal <- merge(highgc, rnaseqnormal, by =
"ensembl_gene_id")
genes_low_gc_cancer <- merge(lowgc, rnaseqcancer, by =
"ensembl_gene_id")
genes_low_gc_normal <- merge(lowgc, rnaseqnormal, by =
"ensembl_gene_id")
```

De esta forma, conseguimos la información para los 10 genes problema con alto %GC y los 10 genes con bajo contenido GC, tanto para células tumorales como para células normales, de sus valores de expresión en cada uno de los experimentos TCGA recogidos en el dataset de RNA-Seq. Para facilitar la identificación de aquellos genes más expresados en células tumorales y en células sanas, calculamos el promedio de los valores de expresión de todos los experimentos TCGA para cada gen problema. Para acabar, exportamos los resultados del análisis en archivos separados.

Para muestras de cáncer:

```
#Media TCGA muestras de cancer:

genes_high_gc_cancer_mean <- transform(
  genes_high_gc_cancer,
  TCGA = rowMeans(genes_high_gc_cancer[,30:1131],na.rm = TRUE)
  )
genes_high_gc_cancer_mean <- genes_high_gc_cancer_mean[,-c(30:1131)]

genes_low_gc_cancer_mean <- transform(
  genes_low_gc_cancer,
  TCGA = rowMeans(genes_low_gc_cancer[,30:1131],na.rm = TRUE)
  )
genes_low_gc_cancer_mean <- genes_low_gc_cancer_mean[,-c(30:1131)]

write.csv(genes_high_gc_cancer_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_high_gc_cancer_mean.csv")

write.csv(genes_low_gc_cancer_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_low_gc_cancer_mean.csv")
```

Para muestras de células sanas:

```
#Media TCGA muestras normales:

genes_high_gc_normal_mean <- transform(
  genes_high_gc_normal,
  TCGA = rowMeans(genes_high_gc_normal[,30:142],na.rm = TRUE)
)
genes_high_gc_normal_mean <- genes_high_gc_normal_mean[,-c(30:142)]

genes_low_gc_normal_mean <- transform(
  genes_low_gc_normal,
  TCGA = rowMeans(genes_low_gc_normal[,30:142],na.rm = TRUE)
)
genes_low_gc_normal_mean <- genes_low_gc_normal_mean[,-c(30:142)]

write.csv(genes_high_gc_normal_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_high_gc_normal_mean.csv")

write.csv(genes_low_gc_normal_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_low_gc_normal_mean.csv")
```

Para los 10 genes con alto contenido GC obtenemos la siguiente información (Tabla 4):

	ensembl_gene_id	Gene	Spectral.Counts.PNNL	Spectral.Counts.VU	percentage_gene_gc_content	TCGA (cancer_mean)	TCGA (normal_mean)
1	ENSG00000111640	GAPDH	13854	9545	60.61	23.821	22.884
2	ENSG00000128591	FLNC	12860	9737	60.86	14.406	16.396
3	ENSG00000142156	COL6A1	7843	7802	64.67	21.286	20.941
4	ENSG00000143632	ACTA1	59482	19259	62.54	13.879	13.188
5	ENSG00000184009	ACTG1	53597	22759	61.09	24.702	24.077
6	ENSG00000188486	H2AFX	8192	5827	66.42	18.896	17.173
7	ENSG00000196924	FLNA	43223	35387	62.28	20.667	21.353
8	ENSG00000197409	HIST1H3D	13036	9197	62.29	15.918	12.591
9	ENSG00000203812	HIST2H2AA3	7978	5269	61.75	10.366	-
10	ENSG00000206172	HBA1	23121	10219	67.26	10.319	12.953

Tabla 4 – Lista de los 10 genes con alto contenido GC seleccionados y con información relevante para su comprensión y comparación: Spectral.Counts de los experimentos PNNL y VU, %GC y valor de expresión de experimentos TCGA para estos genes en muestras tumorales y sanas. Datos extraídos de los archivos `genes_high_gc_cancer_mean.csv` *y* `genes_high_gc_normal_mean.csv`*.*

Para los 10 genes con bajo contenido GC obtenemos la siguiente información (Tabla 5):

	ensembl_gene_id	Gene	Spectral.Counts.PNNL	Spectral.Counts.VU	percentage_gene_gc_content	TCGA (cancer_mean)	TCGA (normal_mean)
1	ENSG00000111799	COL12A1	6625	11461	36.87	19.515	18.409
2	ENSG00000115414	FN1	14726	17853	39.44	22.096	19.342
3	ENSG00000163631	ALB	66797	26718	34.9	12.526	16.073
4	ENSG00000164692	COL1A2	6782	5142	37.57	22.750	21.309
5	ENSG00000166147	FBN1	12992	9741	38.92	18.242	18.770
6	ENSG00000166598	HSP90B1	7882	8569	39.78	22.019	21.682
7	ENSG00000171564	FGB	7603	7476	35.09	11.317	11.615
8	ENSG00000175899	A2M	6885	6568	37.18	20.919	22.349
9	ENSG00000223609	HBD	17234	15031	38.32	9.992	-
10	ENSG00000244734	HBB	34585	21394	37.64	16.000	20.354

Tabla 5 - Lista de los 10 genes con bajo contenido GC seleccionados y con información relevante para su comprensión y comparación: Spectral.Counts de los experimentos PNNL y VU, %GC y valor de expresión de experimentos TCGA para estos genes en muestras tumorales y sanas. Datos extraídos de los archivos `genes_low_gc_cancer_mean.csv` *y* `genes_low_gc_normal_mean.csv`*.*

Datos de expresión génica (F. Iorio)

En este punto, se usaron los datos de expresión génica del trabajo de F. Iorio *et al* (2016) [41] en los que se estudió la expresión de genes en 11.289 tumores de 29 tejidos diferentes. Entre los distintos tipos de cáncer analizados, se estudió el adenocarcinoma de colon y de recto, identificado como COREAD (según la

nomenclatura del TCGA, ver Tabla 1), que es el que en nuestro caso estudiaremos para poder relacionar estos datos con los datos proteómicos del estudio del CPTAC, los cuales también son de cáncer de colon.

Empezamos cargando los paquetes necesarios y la función creada en el apartado anterior para importar datos a R:

```
if(!require(dplyr, xlsx)){
  library(dplyr)
  library(xlsx)
}

loadRData = function(path){
  variable = load(path)
  return(get(variable))
}
```

A continuación, procedemos a cargar el documento .csv con los datos de las muestras (`mm5_samples_by_cancer_type.csv`) del estudio de F. Iorio *et al* [41]. Almacenaremos estos datos bajo la variable `samples`, la cual contiene la información de 1001 experimentos.

```
samples <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/data/mm5_samples_by_cancer_type.csv",
  header = T,
  sep = ";"
  )

str(samples)
```

Seguidamente, seleccionamos únicamente los experimentos (columna "Identifier", detectada mediante la función `str`) asociados a los tumores colorrectales, los cuales en la columna Cancer.Type de samples aparecen como COAD/READ. Por lo tanto, crearemos una variable llamada `samples_coread` que almacene el resultado de filtrar (con `dplyr`) la variable `samples` por el tipo de cáncer COAD/READ.

```
samples_coread <- samples %>% filter(samples$Cancer.Type ==
"COAD/READ")
```

Obtenemos un total de 51 experimentos llevados a cabo con muestras afectadas por cáncer colorrectal.

A continuación, obtenidos del mismo estudio de F. Iorio *et al* (2016) [41], cargaremos los datos de expresión génica (en columnas) asociados a los experimentos mencionados en el dataset anterior ("Identifier" en filas). Almacenaremos los datos en la variable `expression_data`, que contendrá la información de 1018 experimentos ("Identifier").

```
expression_data <- loadRData(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/data/expressionData.Rdata"
  )
```

Realizaremos una serie de transformaciones al dataset `expression_data` para poder llevar a cabo una combinación con el dataset `samples_coread`: convertir en matriz, aislar el nombre de las filas, cambiarles el nombre a "Identifier" (para poder hacer la combinación con `dplyr`) y volver a introducir al dataset original (`expression_data_combo`).

```
expression_data <- as.matrix(expression_data)
expression_data_rows <- as.matrix(rownames(expression_data))
colnames(expression_data_rows) <- "Identifier"
expression_data_combo <- cbind(expression_data, expression_data_rows)
```

Seguidamente, mediante `dplyr`, realizamos una intersección entre los datasets `samples_coread` y `expression_data_combo` en base a la columna común "Identifier" (la etiqueta del experimento).

```
expression_cancer <- merge(samples_coread, expression_data_combo, by =
"Identifier")
```

Obtenemos la información de 49 de los 51 experimentos llevados a cabo en tumores afectados por cáncer colorrectal en el estudio de F. Iorio *et al* (2016) [41] para el estudio de la expresión génica de 17.490 genes.

A continuación, recuperamos el archivo con los genes con alto contenido GC seleccionados anteriormente (`high_gc_genes.csv`).

```
high_gc_genes <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/high_gc_genes.csv",
  header = T
)
```

Una vez conocemos los genes que vamos a tratar, creamos una nueva variable que filtre los nombres de las columnas del dataset `expression_cancer` para verificar si existen los genes problema, en este caso: "GAPDH", "FLNC", "ACTG1", "HIST2H2AA3", "FLNA", "HIST1H3D", "ACTA1", "COL6A1", "H2AFX", "HBA1".

```
expression_data_filter_highgc <- colnames(expression_cancer) %in% c(
"GAPDH","FLNC","ACTG1","HIST2H2AA3","FLNA","HIST1H3D","ACTA1","COL6A1"
,"H2AFX","HBA1")
```

A continuación, creamos la variable `expression_data_highgc` para aplicar el resultado de la consulta anterior sobre el dataset `expression_cancer`. Nos aparecerán los genes que ha encontrado coincidentes con los valores de expresión para cada experimento. Para conocer un valor global, usaremos la función `apply` sobre las columnas de `expression_data_highgc` y convertiremos los datos en numéricos, de forma que podamos aplicar la función `colMeans` para calcular el valor promedio de la expresión de cada gen para cada experimento. Exportamos el resultado en un archivo .csv para su visualización (Tabla 6):

```
expression_data_highgc <-
expression_cancer[expression_data_filter_highgc]
expression_data_highgc <- apply(expression_data_highgc, 2, as.numeric)
expression_data_highgc <-
as.data.frame(colMeans(expression_data_highgc))
colnames(expression_data_highgc) <- "Promedios"
write.csv(expression_data_highgc, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/expression_data_highgc.csv")
```

	Promedios
GAPDH	12,963
FLNC	4,905
COL6A1	4,347
ACTA1	3,374
ACTG1	12,550
H2AFX	11,442
FLNA	5,946
HIST1H3D	3,760

Tabla 6 – Lista de los genes seleccionados con alto contenido GC con los valores de expresión génica globales en base a los experimentos realizados por F. Iorio et al (2016) [41].

En el caso de los genes seleccionados con un bajo contenido GC se llevó a cabo el mismo procedimiento anterior, cambiando únicamente la lista de genes problema (Tabla 7):

```
low_gc_genes <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/low_gc_genes.csv",
  header = T
)

lowgc_names <- as.data.frame(t(data.frame("Genes" =
low_gc_genes$Gene))) #extraiem els noms dels gens

expression_data_filter_lowgc <- colnames(expression_cancer) %in% c(
"ALB","FGB","COL12A1","A2M","COL1A2","HBB","HBD","FBN1","FN1","HSP90B1
")

expression_data_lowgc <-
expression_cancer[expression_data_filter_lowgc]
expression_data_lowgc <- apply(expression_data_lowgc, 2, as.numeric)
expression_data_lowgc <-
as.data.frame(colMeans(expression_data_lowgc))
colnames(expression_data_lowgc) <- "Promedios"
```

	Promedios
COL12A1	3,331
FN1	4,310
ALB	3,090
COL1A2	3,240
FBN1	2,877
HSP90B1	9,787
FGB	3,227
A2M	3,462
HBD	3,179
HBB	3,250

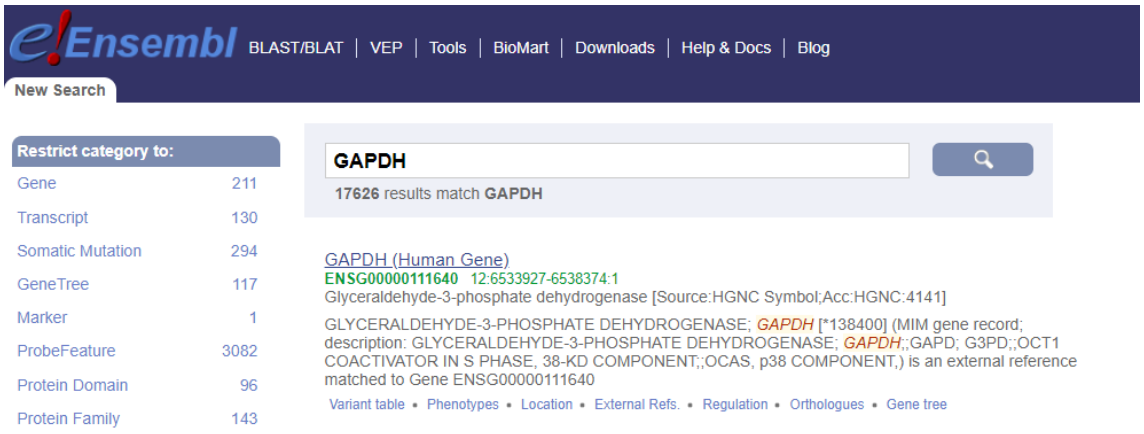
Tabla 7 - Lista de los genes seleccionados con bajo contenido GC con los valores de expresión génica globales en base a los experimentos realizados por F. Iorio et al (2016) [41].

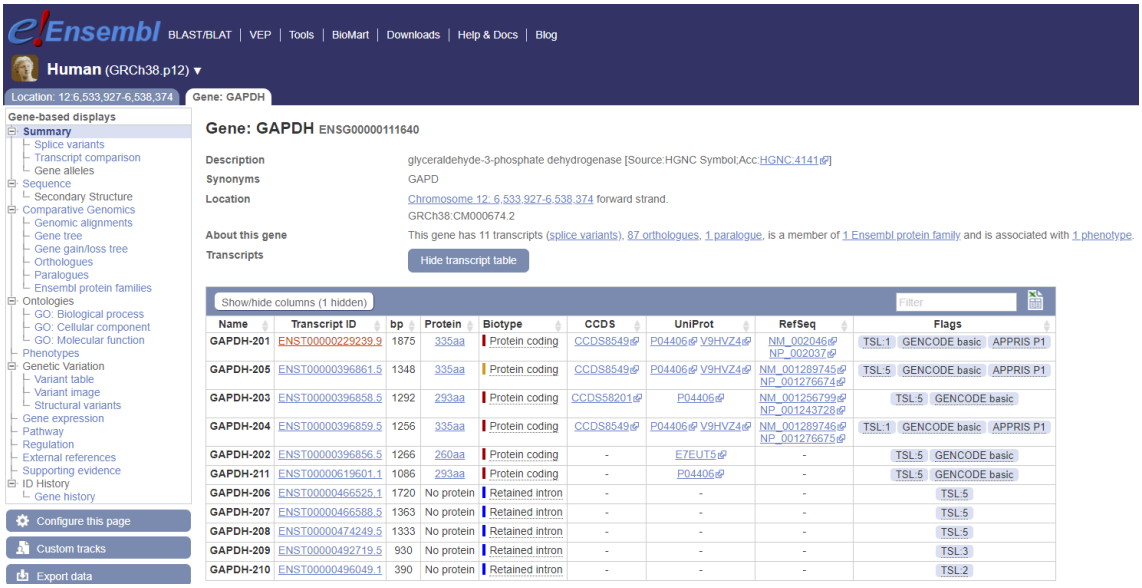
Análisis proteómico

A continuación, para llevar a cabo el estudio a otro nivel más avanzado, llevaremos a cabo la identificación de los péptidos codificados por 4 genes problema altamente expresados en células tumorales y comprobaremos si serían detectables, tanto en su forma wild-type como portando una mutación missense neutral, en los datos proteómicos del estudio del CPTAC.

Para llevar a cabo la selección de los 4 genes problema, usaremos los resultados obtenidos a partir de los datos de expresión génica (F. Iorio *et al* (2016) [41], ver apartado anterior) y los datasets de genes con alto y bajo contenido GC. En el caso de los genes con alto contenido GC, tal y como vemos en la Tabla 6, GAPDH y ACTG1 son los genes con alto contenido GC expresados en cáncer colorrectal con más alto valor de expresión (12,963 y 12,550 respectivamente). Por otra parte, en el caso de los genes con bajo contenido GC, como vemos en la Tabla 7, los genes HSP90B1 y FN1 son los genes con bajo contenido GC, expresados en los experimentos con muestras de cáncer colorrectal F. Iorio *et al* (2016) [41], con los valores más altos de expresión (9,787 y 4,310 respectivamente).

Una vez seleccionados los genes problema para el análisis proteómico, nos dirigiremos a la base de datos de BioMart (Ensembl) y buscaremos la información de cada uno de estos genes en humanos. A continuación, de entre los diferentes transcritos para el mismo gen, seleccionaremos aquel que tenga un mayor tamaño en pares de bases y de lugar a la proteína más grande (mayor número de aminoácidos). Ejemplo para el gen GAPDH a continuación:

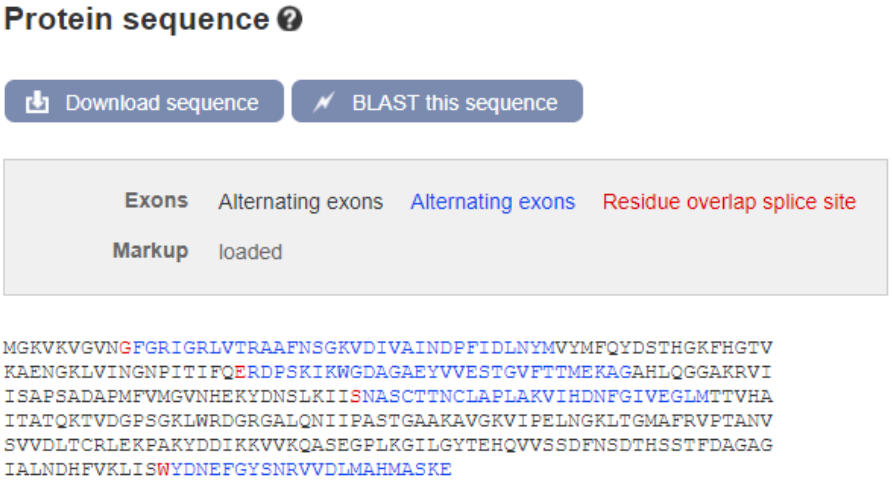




Seleccionamos los siguientes transcritos para cada gen:

- **GAPDH:** ENST00000229239.9
- **ACTG1:** ENST00000575842.5
- **HSP90B1:** ENST00000299767.9
- **FN1:** ENST00000354785.8

Una vez seleccionado el transcrito, visualizamos la secuencia de la proteína codificada del mismo y seleccionaremos dos residuos de Leu o Arg que estén lo más alejados posible (entre 10 y 20 posiciones tanto upstream como downstream) de otros residuos Leu o Arg. De nuevo, ejemplo de visualización de la secuencia de la proteína codificada por el transcrito seleccionado del gen GAPDH:



Para analizar la secuencia con detenimiento, descargaremos la correspondiente a cada gen mediante la opción “Download sequence” en formato FASTA. Una vez visualizada la secuencia con un procesador de texto, como Notepad de Microsoft Windows, buscamos aquellos residuos Leu o Arg codificados por los codones menos comunes de cada aminoácido:

- **Leu**: TTA y CTA
- **Arg**: CGT y CGA

Esta selección es debida a que, en el contexto de una célula tumoral, estos codones podrían llegar a ser vistos en mayor cantidad que en células sanas. No obstante, no en todas las secuencias de los genes problema hemos podido encontrar dos residuos Leu y Arg separados a la distancia mencionada anteriormente. Una vez seleccionado el residuo deseado, seleccionamos la secuencia que va de 10-20 aminoácidos a la izquierda del residuo a 10-20 aminoácidos a la derecha del residuo. Esta será la secuencia wild-type que analizaremos con OpenMS en base a los datos proteómicos del estudio del CPTAC.

Las secuencias wild-type de cada gen problema son las siguientes (en **negrita** el aminoácido seleccionado. **L: Leu, R: Arg**):

>GAPDH_ENST00000229239_TTA_L157L
EKYDNSLKIISNASCTTNC**L**APLAKVIHDNFGIVEGLMTTV

>ACTG1_ENST00000575842_CGT_R177R
TTGIVMDSGDGVTHTVPIYEGYALPHAIL**R**LDLAGRDLTDYLMKILTERGY

>HSP90B1_ENST00000299767_TTA_L61L
EGSRTDDEVVQREEEAIQLDG**L**NASQIRELREKSEKFAFQA

>FN1_ENST00000354785_CGT_R290R
CERHTSVQTTSSGSGPFTDV**R**AAVYQPQPHPQPPPYGHCVTDSGVV

Ahora, para estudiar la aparición de una posible mutación missense neutral en las secuencias peptídicas wild-type de los genes seleccionados, modificaremos el aminoácido señalado por el posible aminoácido surgido a raíz de la mutación missense neutral. Estas serían las secuencias mutantes para cada gen problema que también analizaríamos en OpenMS con los datos proteómicos del CPTAC.

Las secuencias mutantes para cada gen problema son las siguientes (en **negrita** el aminoácido modificado. **V: Val, K: Lys**):

>GAPDH_ENST00000229239_TTA_L157V
EKYDNSLKIISNASCTTNC**V**APLAKVIHDNFGIVEGLMTTV

>ACTG1_ENST00000575842_CGT_R177K
TTGIVMDSGDGVTHTVPIYEGYALPHAIL**K**LDLAGRDLTDYLMKILTERGY

>HSP90B1_ENST00000299767_TTA_L61V
EGSRTDDEVVQREEEAIQLDG**V**NASQIRELREKSEKFAFQA

>FN1_ENST00000354785_CGT_R290K
CERHTSVQTTSSGSGPFTDV**K**AAVYQPQPHPQPPPYGHCVTDSGVV

Una vez seleccionadas estas secuencias, las guardaríamos todas en un fichero de texto con la terminación .fasta para ser analizadas a continuación mediante OpenMS y TOPPAS.

Originalmente, el objetivo era identificar los picos espectrométricos asociados a la secuencia wild-type y a la misma secuencia con una mutación missense neutral. Una vez identificados, cuantificar estos picos para cada muestra y comprobar si hay diferencias significativas, las cuales permitirían distinguir las secuencias según si son wild-type o mutantes.

Para llevar a cabo estos objetivos, el Dr. Miquel Àngel Pujana contactó con el Dr. Eduard Sabidó, líder de la unidad de Proteómica del Centre for Genomic Regulation (CRG), para planificar la estrategia y las posibilidades a seguir para realizar estos análisis mencionados anteriormente. Como ayudante del Dr. Miquel Àngel Pujana, me reuní con el Dr. Eduard Sabidó y con Roger Olivella, informático del grupo del Dr. Eduard Sabidó. Ellos me comentaron que debido a la gran magnitud de datos en el estudio (aproximadamente 170 GB comprimidos), la gran cantidad de muestras (103 para VU, 22 para PNNL a nivel de proteoma, actualmente) y la partición de cada una de las muestras (6 particiones en VU, 12 particiones en PNNL en archivos mzML), teniendo en cuenta además que el grupo no tenía pipelines definidas para este tipo de análisis, se decidió que resultaba inviable a corto plazo realizar el estudio proteómico, tal y como se había planteado inicialmente, para todas las muestras del estudio del CPTAC.

No obstante, se comentó que, en un futuro cercano en el cual yo podría seguir participando, se realizarían los pipelines adaptados para estas muestras para llevar a cabo el análisis para todo el estudio del CPTAC.

A nivel de este Trabajo de Final de Máster, con el objetivo de llevar a cabo una prueba de concepto de un análisis proteómico simple sobre estos datos, se decidió únicamente trabajar con los datos del estudio PNNL, debido a que son menores que los del estudio VU, y solamente identificar la presencia de las secuencias peptídicas wild-type y mutantes, codificadas por los 4 genes problema, en tres muestras al azar, ya que no dispongo del potencial computacional necesario para procesar y analizar simultáneamente los 12 archivos mzML de cada una de las 22 muestras del estudio PNNL.

De esta forma se buscó verificar si la expresión de estos 4 genes problema, expresados en células tumorales de cáncer colorrectal, da lugar a proteínas detectables en el estudio CPTAC, pudiendo además identificar la secuencia mutada.

Para llevar a cabo esta tarea, Roger Olivella (CRG) me cedió un workflow realizado con TOPPAS 2.0 (OpenMS 2.0) para llevar a cabo esta identificación de las secuencias peptídicas problema (Figura 4). El pipeline se compone de:

- **Nodo 1:** introducción de los archivos del análisis espectrométrico en formato estandarizado mzML.
- **Nodo 2:** introducción de la secuencia FASTA que empleará el módulo PeptideIndexer para añadir la información target/decoy para cada péptido analizado. La estrategia target/decoy se basa en que las secuencias necesariamente incorrectas (llamadas decoy) añadidas a la búsqueda implican resultados de búsqueda incorrectos, que de otra forma podrían pasar inadvertidos como correctos, lo cual sería un falso positivo [68].
- **Nodo 3:** introducción de la secuencia FASTA que se usará como base de datos proteómicos de referencia (incluye las secuencias peptídicas a buscar).
- **Nodo 4:** módulo XTandemAdapter, el cual permite la identificación de péptidos en espectros MS/MS mediante el sistema de búsqueda X! Tandem [69], [70].

- **Nodo 5:** módulo IDMerger, el cual mezcla diversos archivos idXML en uno sólo [71].
- **Nodo 6:** módulo PeptideIndexer, que actualiza las referencias proteicas para todas las identificaciones peptídicas (término llamado peptide hits) de un archivo idXML y añade la información target/decoy (mencionado anteriormente).
- **Nodo 7:** módulo FalseDiscoveryRate, el cual permite calcular la ratio de falsos positivos (el número de falsos positivos dividido entre el número total de descubrimientos, dando un valor por encima de un valor de corrección, como podría ser 0.05) [72].
- **Nodo 8:** módulo IDFilter, que como indica su nombre, filtra los resultados de la identificación péptido/proteína según diferentes criterios [73].
- **Nodo 9:** resultados obtenidos del análisis proteico (outputs).

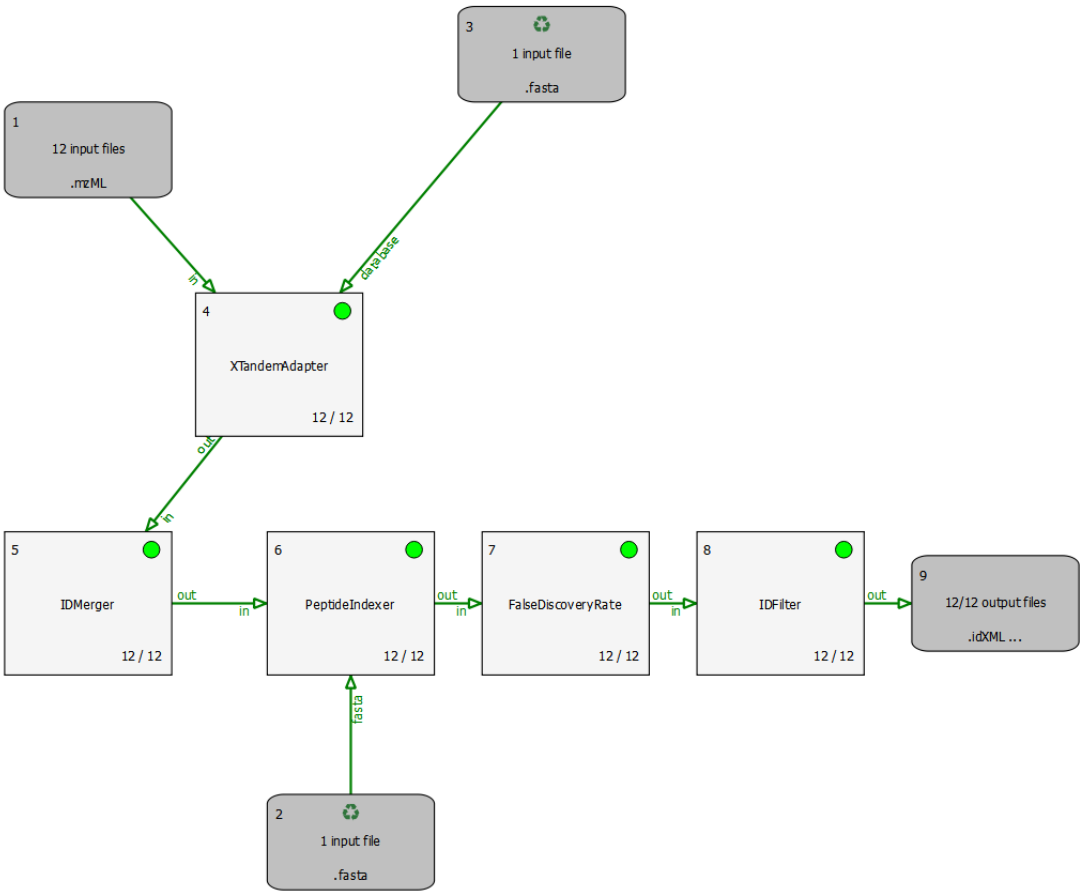


Figura 4 – Workflow para la identificación de las secuencias peptídicas producidas por los genes problema seleccionados. Diseñado por el Dr. Eduard Sabidó y Roger Olivella, CRG.

Una vez se comprobó que el pipeline era funcional para un solo archivo mzML, se pasó a analizar todos los datos mzML simultáneamente (se tuvo que activar la opción Toggle Recycling Mode en el nodo 3 para que se reinicie para cada archivo mzML añadido) por cada una de las muestras y se identificó la presencia o ausencia de las secuencias peptídicas wild-type y mutantes de las proteínas codificadas por los genes problema. En este punto se tenía que tener en cuenta las modificaciones postraduccionales que aplica el workflow a las secuencias peptídicas. Estas serían las mismas secuencias peptídicas anteriores, pero con las modificaciones postraduccionales incluidas:

```
>GAPDH_ENST00000229239_TTA_L157L
IISNASC(Carbamidomethyl)TTNC(Carbamidomethyl)LAPLAK

>GAPDH_ENST00000229239_TTA_L157V
IISNASC(Carbamidomethyl)TTNC(Carbamidomethyl)VAPLAK

>ACTG1_ENST00000575842_CGT_R177R
TTGIVMDSGDGVTHTVPIYEGYALPHAILR

>ACTG1_ENST00000575842_CGT_R177K
TTGIVM(Oxidation)DSGDGVTHTVPIYEGYALPHAILK

>HSP90B1_ENST00000299767_TTA_L61L
EEEAIQLDGLNASQIR

>HSP90B1_ENST00000299767_TTA_L61V
EEEAIQLDGVNASQIR

>FN1_ENST00000354785_CGT_R290R
HTSVQTTSSGSGPFTDVR

>FN1_ENST00000354785_CGT_R290K
HTSVQTTSSGSGPFTDVK
```

A continuación, se intentó crear un script en R que permitiera encontrar cada una de estas secuencias en los archivos mzML resultantes del análisis anterior con OpenMS. Al tratarse de archivos de tipo XML (idXML o IdentityXML), los comandos que hasta ahora he utilizado para el filtraje de datos no pueden ser usados, ya que no es un archivo csv o txt. Utilizando el paquete `XML` de R se intentó realizar un parsing de los archivos XML para leer y extraer la información de los mismos, pero no se consiguió aplicar los recursos online de referencia a este caso particular.

Para llevar a cabo de una forma manual el proceso de identificación de secuencias peptídicas wild-type y mutantes asociadas a los 4 genes seleccionados, se volcaron los 12 archivos XML que componen una misma muestra del CPTAC en un procesador de texto (Notepad) y mediante la opción de búsqueda de caracteres del mismo programa se realizó la búsqueda de las secuencias peptídicas problema. Este proceso se repitió para las tres muestras del estudio PNNL (CPTAC) seleccionadas al azar. El resultado se puede visualizar en la Tabla 8.

Como podemos comprobar, las secuencias peptídicas wild-type codificadas por los genes ACTG1, HSP90B1 y FN1 fueron detectables en repetidas ocasiones en las 3 muestras seleccionadas (aunque la secuencia wild-type de FN1 no se detectó en la muestra 15 del CPTAC). No obstante, ni la secuencia wild-type codificada por el gen GAPDH ni las secuencias mutadas con una mutación missense neutral asociadas a los 4 genes problema fueron detectables en las muestras estudiadas.

Muestra	Gen	Presencia (SI/NO)		Muestra	Gen	Presencia (SI/NO)		Muestra	Gen	Presencia (SI/NO)
CPTAC 10 (PNNL)	GAPDH wild-type	NO		**CPTAC 15 (PNNL)**	GAPDH wild-type	NO		**CPTAC 20 (PNNL)**	GAPDH wild-type	NO
	GAPDH mutado	NO			GAPDH mutado	NO			GAPDH mutado	NO
	ACTG1 wild-type	SI (4 HITS)			ACTG1 wild-type	SI (4 HITS)			ACTG1 wild-type	SI (8 HITS)
	ACTG1 mutado	NO			ACTG1 mutado	NO			ACTG1 mutado	NO
	HSP90B1 wild-type	SI (3 HITS)			HSP90B1 wild-type	SI (3 HITS)			HSP90B1 wild-type	SI (7 HITS)
	HSP90B1 mutado	NO			HSP90B1 mutado	NO			HSP90B1 mutado	NO
	FN1 wild-type	SI (2 HITS)			FN1 wild-type	NO			FN1 wild-type	SI (3 HITS)
	FN1 mutado	NO			FN1 mutado	NO			FN1 mutado	NO

Tabla 8 – Identificación de las secuencias peptídicas wild-type y mutantes asociadas a los genes problema seleccionados (GAPDH, ACTG1, HSP90B1 y FN1) en las muestras del ensayo del Pacific Northwest National Laboratory (PNNL) dentro del estudio del CPTAC Cancer Proteome Confirmatory Colon Study.

Discusión

En este trabajo se ha intentado identificar la presencia de mutaciones missense neutrales en codones codificantes de leucina (Leu) y arginina (Arg) mediante el análisis de datos de genómicos, transcriptómicos y proteómicos de distintas bases de datos y estudios. Con esta identificación, se pretendía inicialmente poder demostrar que la generación de mutaciones missense neutrales se incrementa en células tumorales respecto a células sanas.

Partimos inicialmente de un dataset de 4685 genes y 4622 genes identificados respectivamente en los ensayos del Pacific Northwest National Laboratory (PNNL) y la Universidad de Vanderbilt (VU), los dos miembros que llevaron a cabo el estudio del CPTAC Cancer Proteome Confirmatory Colon Study entre el año 2015 y 2017.

Dado que el paso final del estudio es la identificación de las secuencias peptídicas wild-type y mutantes de unos cuantos genes seleccionados, necesitamos filtrar esta ingente cantidad de genes para quedarnos únicamente con aquellos que tengan un espectro mínimamente visible y cuantificable. Para ello, seleccionamos aquellos con un valor de Spectral.Counts (total de espectros identificados para una misma proteína) superior o igual a 100. Como resultado, obtenemos 2805 genes para PNNL y 2824 genes para VU, reduciendo aproximadamente un 50% el total de genes a analizar.

A continuación, se mezclaron ambos datasets (PNNL y VU) para solo seleccionar aquellos genes que hubieran sido analizados por ambos grupos. En total, 2198 genes se seleccionaron.

Sabiendo ahora con que tamaño de genes trabajamos, necesitamos conocer las propiedades intrínsecas de cada uno de ellos. Para llevar a cabo esta tarea, hacemos uso del paquete Bioconductor en R para extraer la información relativa a *Homo sapiens* de cada uno de los 2198 genes anteriores a través de BioMart (Ensembl).

Seguidamente, para seguir filtrando los genes problema y determinar si el porcentaje de CG en los genes tiene un impacto en los niveles de transcripción y en la traducción de estos, filtramos ahora los 10 genes con más contenido GC (valor de Spectral.Counts superior a 7500 y un contenido en GC igual o superior al

60%) y los 10 con menor porcentaje de GC (valor Spectral.Counts superior a 6600 y un contenido en GC menor o igual a un 40%). De esta forma, en un análisis proteómico con los datos del CPTAC, cabría la posibilidad que los genes con mayor proporción de GC mostraran mayores niveles de traducción y mayor cantidad de proteína, mientras que los genes con menor porcentaje de GC podrían mostrar menores niveles de traducción y menor cantidad de proteína producida.

Una vez seleccionados los genes según su contenido GC, usaremos los datos (normalizados previamente por el grupo del Dr. Miquel Àngel Pujana) de RNA-Seq de células cancerosas y sanas obtenidos del GDC (Genomic Data Commons), dentro del TCGA. En este caso, los datos de expresión en RNA-Seq derivan de células cancerosas afectadas por mutaciones en los genes BRCA, las cuales carecen de la capacidad de producir proteínas supresoras de tumores funcionales generadas por BRCA1 y BRCA2 en condiciones normales, impidiendo así la reparación correcta del DNA dañado e incrementando en consecuencia las posibilidades de presentar mutaciones que conduzcan a un cáncer [74].

Como se ha podido ver a lo largo del trabajo, el objetivo de este estudio es la detección y cuantificación de mutaciones missense neutrales de Leu y Arg en genes expresados en células de cáncer colorrectal y células sanas. En este caso, usamos los datos de RNA-Seq para reforzar la consistencia de los resultados encontrados y debido a que los genes seleccionados para el análisis proteómico son expresados en múltiples tipos celulares, por lo que se podría deducir que en cáncer colorrectal también deberían expresarse en mayor o menor medida. Los genes seleccionados son (información extraída de NCBI Gene y GeneCards para *Homo sapiens*):

- **GAPDH:** gliceraldehido-3-fosfato deshidrogenasa. La proteína generada por este gen tiene implicaciones directas en la glucólisis y en diversas funciones nucleares, como la nitrosilación de proteínas. Se expresa ubicuamente en el cuerpo humano, concretamente en 27 tejidos, entre ellos el colon [75], [76].
- **ACTG1:** actina gamma 1. Este gen codifica para isoforma gamma de la actina, una proteína globular altamente conservada que tiene un rol en diversos tipos de motilidad celular y en el mantenimiento del

citoesqueleto. En este caso, la isoforma gamma de la actina se encuentra en forma citoplasmática en múltiples tipos celulares, especialmente en el colon [75], [76].

- **HSP90B1:** heat shock protein 90 beta family member 1. Este gen genera un miembro de la familia de chaperonas que llevan a cabo la metabolización del ATP (adenosina trifosfato), así como la estabilización y el plegamiento de otros tipos de proteínas. La expresión de esta proteína de choque térmico se asocia a una gran variedad de estados patogénicos, como la tumorigénesis o formación de tumores. Como las proteínas anteriores, la HSP90B1 se expresa de forma generalizada por todo el organismo humano, incluyendo el colon [75], [76].
- **FN1:** fibronectina 1. FN1 codifica la proteína fibronectina, una glicoproteína presente en forma de dímero en el plasma y en forma dimérica o multimérica en la superficie de la matriz extracelular. Esta proteína tiene un papel relevante en la adhesión celular, la embriogénesis, la coagulación sanguínea, el sistema inmune y en la metástasis. Aunque se expresa principalmente en la placenta, esta proteína también se encuentra expresada en el colon [75], [76].

Dado que estos 4 genes seleccionados en el estudio se expresan en el colon, podemos suponer que se expresaran en una situación de cáncer colorrectal como la que se trata en este estudio.

Una vez exportados los datasets de RNA-Seq para células normales y cancerosas, filtramos los genes estudiados en los múltiples experimentos TCGA con los genes con alto y bajo contenido GC seleccionados anteriormente. Seguidamente, se aplica el promedio de los valores de expresión de cada gen por cada experimento TCGA llevado a cabo en células sanas y tumorales. Este resultado aparece representado en la Tabla 4 y 5.

En ambos casos, vemos que hay escasa diferencia entre la expresión en células de cáncer y en células sanas. Además, vemos que no existe un patrón en base a un incremento de la expresión en situación de cáncer, ya que algunos genes se ven más expresados en células normales y a la inversa. Por otra parte, vemos que en la Tabla 4 el gen HIST2H2AA3 (Histone Cluster 2 H2A Family Member A3), codificante de una histona de la familia H2A, únicamente se expresa en

células tumorales, no se ha detectado expresión en células normales. De forma similar, en la Tabla 5, vemos como el gen HBD (Hemoglobin Subunit Delta), que codifica para las subunidades gamma de la hemoglobina, solo se expresa en células cancerosas. Es un dato curioso porque, en ambos casos, se trata de genes que codifican para proteínas ubicuamente presentes en el organismo humano, como son las histonas, que permiten la compactación del DNA, o la hemoglobina, encargada de transportar el oxígeno molecular desde los órganos respiratorios a los diferentes tejidos, así como de transportar el CO2 de estos hacia los pulmones para su expulsión. No obstante, es cierto que la forma de hemoglobina que forma las subunidades gamma (HbA-2) solo forma el 3% del total de hemoglobina humana (junto con HbF), lo cual podría explicar que quizá no fuera detectado este gen dada su baja presencia en el organismo [75], [76].

Para comprobar la significancia estadística de los datos de expresión génica, podemos realizar un test estadístico como el test t de Student para comprobar si existen diferencias significativas entre los valores de expresión en células cancerosas y normales. No obstante, antes de realizar este test debemos asegurarnos que los niveles de expresión en células cancerosas y tumorales siguen una distribución normal en forma de campana de Gauss. Para verificar la normalidad de los valores realizamos un test Shapiro-Wilk:

```
shapiro.test(genes_high_gc_cancer_mean$TCGA)
shapiro.test(genes_high_gc_normal_mean$TCGA)
shapiro.test(genes_low_gc_cancer_mean$TCGA)
shapiro.test(genes_low_gc_normal_mean$TCGA)
```

Los p-values obtenidos en cada caso son: 0.4938, 0.2234, 0.1598 y 0.1647. Si suponemos que la hipótesis nula es que todos los datos provienen de una distribución normal, dado que todos los p-values superan el valor de significación del 0.05, podemos aceptar la hipótesis nula y afirmaríamos que los niveles de expresión se distribuyen siguiendo una distribución normal.

A continuación, para determinar la homocedasticidad (igualdad de las varianzas entre conjuntos de datos) entre los niveles de expresión de ambas clases de células (tumorales y sanas), realizaremos un test F de Fisher:

```
var.test(genes_high_gc_cancer_mean$TCGA,
genes_high_gc_normal_mean$TCGA)
```

```
var.test(genes_low_gc_cancer_mean$TCGA, genes_low_gc_normal_mean$TCGA)
```

En el caso de los genes con alto %GC, obtenemos un p-value de 0.6829, mientras que para los genes con bajo contenido GC se obtiene un p-value de 0.3202. Si suponemos que la hipótesis nula es que las dos muestras presentan varianzas iguales, dado que en ambos casos los p-values son superiores al nivel de significación de 0.05 aceptaríamos la hipótesis nula, afirmando que las varianzas entre los niveles de expresión de células tumorales y sanas son iguales.

Dado que sabemos que los niveles de expresión siguen una distribución normal y entre tipos celulares tienen la misma varianza, podemos aplicar el siguiente código para realizar el test t de Student entre las dos últimas columnas de la Tabla 4 (el mismo procedimiento se sigue para la Tabla 5, correspondiente a los 10 genes seleccionados con bajo porcentaje GC):

```
ttest_highGC <- t.test(genes_high_gc_cancer_mean$TCGA,
genes_high_gc_normal_mean$TCGA, var.equal = TRUE)
ttest_highGC

ttest_lowGC <- t.test(genes_low_gc_cancer_mean$TCGA,
genes_low_gc_normal_mean$TCGA, var.equal = TRUE)
ttest_lowGC
```

El resultado del test t de Student para los 10 genes con alto %GC son:

```
Two Sample t-test

data:  genes_high_gc_cancer_mean$TCGA and
genes_high_gc_normal_mean$TCGA
t = -0.2333, df = 17, p-value = 0.8183
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.271035  4.221395
sample estimates:
mean of x mean of y
 17.42634  17.95116
```

En este caso, si suponemos como hipótesis nula que los niveles de expresión en células cancerosas y células sanas presentan medias similares y no tienen diferencias significativas, dado que el p-value obtenido es del 0.8183, por lo tanto, muy por encima del nivel de significación del 0.05, aceptaríamos la hipótesis nula y afirmaríamos que no existen diferencias significativas entre los niveles de expresión en ambos tipos de células. Además, si nos fijamos en los

valores promedios de ambos niveles de expresión, vemos que no hay prácticamente diferencias entre la expresión de los genes problema en células cancerosas (17.42634) y la expresión de los mismos genes en células sanas (17.95116), es decir, una diferencia cercana al 3% ($(\frac{17.95116 - 17.42634}{\frac{17.95116 + 17.42634}{2}}) \cdot 100 = 2{,}967\%$).

Obtenemos el siguiente resultado al aplicar el test t de Student para los 10 genes con bajo porcentaje de GC:

```
Two Sample t-test

data:  genes_low_gc_cancer_mean$TCGA and genes_low_gc_normal_mean$TCGA
t = -0.69918, df = 17, p-value = 0.4939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.385955  2.704760
sample estimates:
mean of x mean of y
 17.53809  18.87869
```

Si aceptamos la hipótesis nula de que no existen diferencias significativas entre los niveles de expresión de los genes problema en células tumorales y sanas, obteniendo un p-value de 0.4939, superior al nivel de significación del 0.05, podemos aceptar la hipótesis nula y afirmar que, en efecto, no existen diferencias significativas entre los niveles de expresión entre células cancerosas y sanas. Además, como se ha podido comprobar anteriormente, los valores promedio de los niveles de expresión en ambas poblaciones celulares son muy parecidas: 17.53809 para células cancerosas y 18.87869 para células normales. Dado que existe una diferencia ligeramente superior del 7% (7,363%), podemos reforzar la afirmación de que no existen diferencias significativas entre los niveles de expresión de los genes problema en células tumorales y en células sanas.

Por otra parte, podríamos intentar determinar si existen diferencias significativas entre los niveles de expresión de genes con alto y bajo contenido GC. Dado que hemos comprobado que los datos siguen una distribución normal y entre tipos celulares tienen la misma varianza, podemos aplicar el test t de Student entre los niveles de expresión de genes con alto y bajo %GC en células tumorales y realizar el mismo test para células normales.

En el caso de células cancerosas obtenemos el siguiente resultado:

```
test_cancer <- t.test(genes_high_gc_cancer_mean$TCGA,
genes_low_gc_cancer_mean$TCGA, var.equal = TRUE)
test_cancer

      Two Sample t-test

data:  genes_high_gc_cancer_mean$TCGA and
genes_low_gc_cancer_mean$TCGA
t = -0.04983, df = 18, p-value = 0.9608
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.823680  4.600162
sample estimates:
mean of x mean of y
 17.42634  17.53809
```

Podemos comprobar que el p-value obtenido (0.9608) es muy superior al nivel de significación de 0.05. Si suponemos la hipótesis nula de que no existen diferencias significativas entre los niveles de expresión de los genes con alto %GC y los genes con bajo contenido GC en células cancerosas, dado el p-value obtenido, deberíamos aceptar la hipótesis nula y afirmar que no existen diferencias significativas en células cancerosas entre los niveles de expresión de genes con alto porcentaje GC y genes con bajo contenido GC.

Por otra parte, en el caso de células sanas:

```
test_normal <- t.test(genes_high_gc_normal_mean$TCGA,
genes_low_gc_normal_mean$TCGA, var.equal = TRUE)
test_normal

      Two Sample t-test

data:  genes_high_gc_normal_mean$TCGA and
genes_low_gc_normal_mean$TCGA
t = -0.49692, df = 16, p-value = 0.626
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.884496  3.029422
sample estimates:
mean of x mean of y
 17.95116  18.87869
```

En este caso obtenemos un p-value de 0.626, también muy superior al nivel de significación de 0.05, por lo que en este caso también deberíamos aceptar la hipótesis nula, la cual implica que no hay diferencias significativas en células sanas entre los niveles de expresión de genes con alto %GC y genes con bajo %GC. Por lo tanto, parece ser que la hipótesis que implicaba que los genes con

alto contenido en GC se expresaban más que aquellos con bajo porcentaje de GC no se cumpliría ni en células cancerosas ni en células sanas.

Una vez conocemos los niveles de expresión de los genes problema en células tumorales y en células sanas y hemos comprobado que no existen diferencias significativas entre ellos, emplearemos los datos de expresión génica de F. Iorio *et al* (2016) [41] para comprobar la expresión en otro contexto diferente. De los 11.289 tumores estudiados en 29 tejidos distintos, se seleccionaron únicamente los experimentos relacionados con el adenocarcinoma de colon y de recto, etiquetado como COREAD, ya que es nuestro objeto de estudio identificar las secuencias peptídicas wild-type y mutantes de los genes problema usando los datos proteómicos del estudio de cáncer colorrectal CPTAC. Una vez filtradas las muestras asociadas al cáncer colorrectal, se carga el dataset de expresión génica realizado por el mismo grupo en relación a los experimentos llevados a cabo en el estudio. A continuación, filtramos este dataset de expresión para solo obtener los datos de las muestras asociadas a cáncer colorrectal, obteniendo así los niveles de expresión de 17.490 genes en 49 experimentos de cáncer colorrectal. Seguidamente, filtraremos este último dataset con el nombre de los genes problema seleccionados según su alto/bajo contenido en GC, se almacenan los resultados para los 10 genes con alto %GC y los 10 genes con bajo %GC por separado y se calcula el promedio de los valores de expresión de todos los experimentos realizados donde se estudió cada gen. Estos resultados aparecen representados en la Tabla 6 y 7.

Como podemos comprobar en la Tabla 6, sólo se ha podido encontrar la expresión de 8 genes de los 10 seleccionados con alto contenido GC en este estudio. En este caso, vemos que los genes HIST2H2AA3 y HBA1, los cuales si podíamos ver expresados con los datos de RNA-Seq del GDC (ver Tabla 4), no se expresan en ninguno de los 49 experimentos llevados a cabo con muestras de cáncer colorrectal. De todas formas, si nos fijamos en los valores de expresión de HIST2H2AA3 y HBA1 en la Tabla 4, vemos que eran los dos genes que menos se expresaban tanto en células tumorales (HIST2H2AA3: 10.366) como en células tumorales y sanas (HBA1: 10.319 para células cancerosas, 12.953 para células sanas). Posiblemente, el estudio de F. Iorio *et al* (2016) [41] no fue capaz de detectar niveles de expresión tan bajos comparados con los de ACTG1

(24.702 en células de cáncer, 24.077 en células normales en los datos de RNA-Seq del GDC).

Esta hipótesis podría ser posible si no fuera por el caso de la Tabla 7, donde aquí si aparecen los datos de expresión de los 10 genes con bajo contenido GC seleccionados, a pesar de que genes como HBD, en el dataset de RNA-Seq anterior, mostraban unos niveles de expresión en células tumorales de 9.992 (ver Tabla 5), inferiores a los comprobados para genes como HIST2H2AA3 o HBA1, que no fueron captados en este estudio de F. Iorio *et al* (2016) [41].

Dada esta situación, cabe la posibilidad que sencillamente los genes con alto %GC como son HIST2H2AA3 y HBA1 no fueran detectados o analizados en el estudio de F. Iorio *et al* (2016) [41], independientemente de los niveles de expresión que estos mostraran.

Por otra parte, en este punto podríamos determinar si los genes con un elevado porcentaje de GC se expresan en mayor medida que aquellos con un contenido GC menor, demostrando así la hipótesis que indica que el porcentaje de GC es un indicador de los niveles de transcripción del gen, pudiendo afectar así la eficiencia de la traducción de los transcritos del gen en cuestión. Esta hipótesis fue rechazada con los valores de expresión RNA-Seq del GDC tanto para células cancerosas como sanas (valores mostrados en la Tabla 4 y 5), posiblemente debido a que los genes se expresaban en un rango similar de valores independientemente del contenido GC. No obstante, esto no está tan claro en las Tablas 6 y 7, donde sí se pueden comprobar diferencias sustanciales entre los rangos de expresión génica de los genes con alto o bajo contenido GC.

De forma similar al análisis estadístico realizado con los datos de expresión génica obtenidos del dataset de F. Iorio *et al* (2016) [41], en este caso (datos RNA-Seq del GDC) también podríamos realizar una prueba t de Student entre los valores de expresión totales de los genes con alto contenido GC y bajo porcentaje de GC para determinar si existen diferencias significativas entre ellos. En primer lugar, determinamos la normalidad de los datos con un test Shapiro-Wilk:

```
shapiro.test(expression_data_highgc$Promedios)
shapiro.test(expression_data_lowgc$Promedios)
```

Para los genes con alto contenido GC obtenemos un p-value de 0.0347, mientras que para los genes con bajo contenido GC tenemos un p-value de $4.492e^{-06}$. Por

lo tanto, si suponemos que la hipótesis nula es que los datos provienen de una distribución normal, y dado que los p-values obtenidos son menores que el valor de significación de 0.05, podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa, de forma que afirmamos que los valores de expresión génica para ambos grupos de genes no se distribuyen de forma normal. Esto puede ser debido a que dentro de los mismos grupos de genes tenemos rangos de expresión genética muy amplios, con genes con una expresión ligeramente superior a 3 a genes con una expresión superior a 12. Esta falta de uniformidad en los valores quizá podría explicar la falta de distribución normal.

Dado que el tamaño de los datos no es suficientemente grande en ninguno de los dos casos, no podemos suponer el teorema del límite central, dado que al no disponer de una gran cantidad de datos estos no pueden simular o aproximarse a una distribución normal. Dado que no podemos usar el test t de Student debido a la falta de normalidad, podemos utilizar el test de Mann-Whitney U, el cual es no paramétrico (no asume que los datos siguen una distribución específica). De forma similar al test t de Student, para poder aplicar el test de Mann-Whitney las varianzas en los dos grupos deben ser iguales. Uno de los test con los que podemos comprobar esta suposición es con el test de Levene, en el cual la hipótesis nula afirma que las varianzas son iguales (homocedasticidad). En el código siguiente, agruparíamos los datos de expresión promedios de cada grupo en un mismo vector (`grupomuestras`) y creamos otro vector que permita definir ambos grupos en forma de factores (`grupo`). Finalmente, aplicamos el test de Levene:

```
grupomuestras <- c(expression_data_highgc$Promedios,
expression_data_lowgc$Promedios)
grupo <- as.factor(c(rep(1, length(expression_data_highgc$Promedios)),
rep(2, length(expression_data_lowgc$Promedios))))
library(Rcmdr)
leveneTest(grupomuestras, grupo)
```

Obtenemos el siguiente resultado:

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value  Pr(>F)
group  1  4.2474 0.05595 .
      16
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

En este caso, podemos comprobar que curiosamente el p-value es de 0.05595, por lo tanto, ya que supera ligeramente el nivel de significación de 0.05, podemos aceptar la hipótesis nula y afirmar que las varianzas son iguales.

Una vez comprobadas que las varianzas son iguales podemos proceder a realizar el test de Mann-Whitney U (conocido también como test de Mann-Whitney-Wilcoxon):

```
wilcox.test(expression_data_lowgc$Promedios,
expression_data_highgc$Promedios)
```

Obtenemos el siguiente resultado:

```
Wilcoxon rank sum test

data:  expression_data_lowgc$Promedios and
expression_data_highgc$Promedios
W = 8, p-value = 0.003062
alternative hypothesis: true location shift is not equal to 0
```

Como podemos comprobar, el p-value de 0.003062 está muy por debajo del nivel de significación de 0.05, de forma que se rechazaría la hipótesis nula (ambos promedios son iguales) y se aceptaría hipótesis alternativa, que implicaría que hay diferencias significativas entre los valores de expresión génica promedio de los genes con alto contenido GC y los genes con bajo %GC, lo cual implicaría que existen evidencias para confirmar que los genes con alto contenido GC se expresan en mayor medida (valor promedio de los niveles de expresión de los genes problema de 7.410953, `mean(expression_data_highgc$Promedios)`) que los genes con bajo contenido GC (valor promedio de la expresión de los genes seleccionados de 3.975386, `mean(expression_data_lowgc$Promedios)`).

Una vez conocemos los niveles de expresión de los genes problema en células de cáncer colorrectal, seleccionamos los 2 genes con alto %GC más expresados (GAPDH y ACTG1) y los 2 genes con bajo contenido GC también más expresados (HSP90B1 y FN1). Seguidamente, obtenemos las secuencias peptídicas de los transcritos de cada gen que den lugar a las proteínas con mayor nombre de aminoácidos. A continuación, mediante un procesador de texto como Notepad seleccionamos los residuos Leu y Arg codificados por sus codones menos comunes (TTA y CTA para Leu, CGT y CGA para Arg) y separados de otros aminoácidos Leu y Arg en 10-20 posiciones upstream y downstream. Tal y

como se mencionaba en la sección correspondiente al análisis proteómico en **Resultados**, creemos que en células tumorales la presencia de estos codones poco comunes para cada residuo podría ser mayor, dada las circunstancias anómalas de la célula en cuestión.

Una vez seleccionadas las secuencias de aminoácidos que contienen el residuo de interés para cada gen (secuencias wild-type), modificaríamos el aminoácido problema como si se tratase de una mutación missense neutral, la cual en el caso de la Leu esta se convertiría en una Val mientras que el residuo de Arg se convertiría en un residuo de Lys. Estas secuencias modificadas por nosotros serían las secuencias mutantes de cada proteína problema.

A continuación, volcaríamos todas las secuencias en un archivo .fasta previamente parametrizado por Roger Olivella (CRG) con el fin de ser usadas para la identificación de los péptidos wild-type y mutantes codificados por los 4 genes problema seleccionados en las muestras del estudio del CPTAC mediante el software OpenMS y su aplicación, TOPPAS. Cabe mencionar, así como se comentó en su apartado correspondiente en la sección de **Resultados**, que, debido a las limitaciones relacionadas con la magnitud de los datos del estudio, la gran cantidad de muestras, el fraccionamiento de los archivos de las muestras y la falta de pipelines efectivos para el estudio, resultaba inviable a corto plazo llevar a cabo el objetivo original de identificar y cuantificar los picos espectrométricos correspondientes a los péptidos wild-type y mutantes en cada muestra del estudio, con el fin de encontrar diferencias relevantes entre ellos.

Como parte de este Trabajo de Final de Máster, como objetivo más simple y alcanzable, se llevó a cabo la identificación de las secuencias peptídicas problema (wild-type y mutante) de los 4 genes seleccionados únicamente usando las muestras correspondientes al estudio del PNNL, ya que había una menor cantidad de estas respecto a las estudiadas por VU. De las 22 muestras del estudio de PNNL, se seleccionaron 3 muestras al azar (la 10, 15 y 20), ya que carezco del potencial computacional necesario para estudiar todas las muestras simultáneamente.

Mediante el uso del software TOPPAS 2.0 y del workflow diseñado por Roger Olivella, se procesaron y analizaron todas las secuencias peptídicas contenidas

en las 3 muestras seleccionadas al azar. A pesar de que se intentó crear un script en R que permitiera la identificación automatizada de las secuencias peptídicas wild-type y mutantes de los genes problema en los archivos idXML generados por TOPPAS, fui incapaz de emplear los recursos online de referencia sobre parsing en archivos XML con los archivos idXML.

De forma alternativa, volqué los archivos idXML correspondientes a cada muestra en archivos de texto por separado y, empleando la herramienta interna de búsqueda de caracteres del propio Notepad, llevé a cabo la identificación de las secuencias problema.

Como se puede comprobar en la Tabla 8, las secuencias peptídicas wild-type asociadas a los genes ACTG1, HSP90B1 y FN1 fueron detectadas múltiples veces en las 3 muestras seleccionadas (aunque la secuencia wild-type de FN1 no se detectó en la muestra 15 del CPTAC). Por otra parte, ni la secuencia peptídica wild-type codificada por GAPDH ni ninguna de las secuencias peptídicas mutantes para cada uno de los 4 genes seleccionados fueron identificadas en las 3 muestras aleatorias escogidas del estudio de PNNL.

Por lo tanto, a pesar de que no se pudo realizar el análisis proteómico como se había planteado inicialmente, se ha podido realizar una prueba de concepto, en que, partiendo de diversos datasets de datos ómicos para células tumorales y normales, se pueden seleccionar un set de genes determinado según sus niveles de expresión y verificar la presencia de sus productos proteicos en un dataset de datos proteómicos.

Conclusiones

Mediante la realización de este Trabajo de Final de Máster se ha pretendido ofrecer unas primeros resultados y conclusiones sobre la línea de investigación iniciada por el Dr. Miquel Àngel Pujana para identificar y cuantificar la presencia de mutaciones missense neutrales en el contexto de células cancerosas a partir de datos recogidos de diversas ómicas.

Partiendo de datasets recogidos y tratados previamente por el grupo, como serían los análisis de expresión génica en líneas tumorales, llevados a cabo originalmente por F. Iorio *et al* (2016) [41], los datos de RNA-Seq obtenidos del GDC (TCGA), o la información genómica de cada uno de los genes seleccionados recogida gracias a BioMart, se ha realizado un caracterización y análisis de la expresión génica de una serie de genes seleccionados según su porcentaje de GC, los cuales fueron inicialmente escogidos a partir del estudio proteómico del CPTAC (Colon Cancer Confirmatory Study), llevado a cabo por la Universidad de Vanderbilt y el Pacific Northwest National Laboratory.

Para llevar a cabo estos análisis, como se ha podido comprobar a lo largo de este proyecto, puse a prueba mis habilidades en programación en R, así como mis conocimientos en estadística y en el manejo y gestión de datasets de diversa índole. Dada mi formación en ciencias biológicas, esta parte fue posiblemente una de las más arduas del trabajo, aunque tuve la fortuna de contar con la ayuda de Luís Palomero, bioinformático colaborador del Dr. Miquel Àngel Pujana, que me asistió siempre que tenía alguna duda en este aspecto más computacional del trabajo.

Para dotar al trabajo de un aspecto más completo, decidí incluir en él los análisis de los datos proteómicos del CPTAC. Originalmente se pretendía identificar las secuencias peptídicas de los genes seleccionados en los estudios de la expresión génica y verificar si estas eran wild-type o bien contenían mutaciones missense neutrales en los codones menos comunes de Leu y Arg. Una vez verificadas estas secuencias se procedería a cuantificar los picos espectrométricos de las secuencias wild-type y las mutantes para cada muestra estudiada en el CPTAC y se comprobaría si las diferencias serían significativas y relevantes. De esta forma, se pretendía como objetivo final poder demostrar que estas mutaciones missense neutrales en los codones menos comunes de Leu y Arg sucedían con más frecuencia en células cancerosas que en células sanas normales.

Dado que en el grupo del Dr. Miquel Àngel Pujana no había bioinformáticos con la formación necesaria para manipular y estudiar datos proteómicos ni yo tenía los conocimientos requeridos para ello, el Dr. Miquel Àngel Pujana se puso en contacto con el Dr. Eduard Sabidó, líder de la unidad de Proteómica del CRG, en el Parque de Investigación Biomédica de Barcelona.

Aunque la planificación inicial del trabajo se siguió en las primeras etapas sin desviaciones temporales substanciales, la dificultad de poder establecer una reunión entre los miembros de ambos grupos, la gran cantidad de datos del estudio proteómico del CPTAC (cerca de 170 GB de datos comprimidos) y la baja calidad de mi conexión personal a Internet, produjeron una serie de desviaciones temporales notables.

No obstante, en las dos últimas semanas antes de la entrega de este trabajo pude reunirme con el Dr. Eduard Sabidó y Roger Olivella y discutir cómo podríamos proceder a corto y largo plazo con el análisis de los datos proteómicos del CPTAC. Se decidió que, a largo plazo, se deberían crear pipelines que permitieran procesar los archivos mzML de todos los datos de proteoma de las muestras de VU y PNNL. Una vez procesados, sería necesario categorizar todas las proteínas analizadas en las muestras y asignar los picos espectrométricos a estas. A continuación, se podría cuantificar la intensidad de los picos de las proteínas wild-type, así como de las proteínas mutadas por mutaciones del tipo missense neutral. De esta forma se podría llegar a desarrollar un método que pudiera detectar, de forma automatizada para un número dado de muestras, este tipo de mutaciones a nivel de proteoma mediante la comparación de su pico espectrométrico con el pico espectrométrico de la misma proteína wild-type.

Dado que este objetivo final requiere mucho tiempo y un profundo conocimiento en análisis y procesamiento de datos proteómicos, el Dr. Eduard Sabidó y Roger Olivella me aconsejaron que, a corto plazo y a modo de prueba de concepto, llevara a cabo la identificación de las secuencias peptídicas wild-type y mutantes de los 4 genes seleccionados mediante los análisis de datos transcriptómicos y genómicos. Este análisis, que aparece comentado en el apartado de **Procedimiento y resultados obtenidos** y **Discusión**, me permitió demostrar que, partiendo de unos genes seleccionados a partir de datos ómicos según una serie de parámetros, como podría ser el contenido en GC y/o los niveles de

expresión génica, se puede encontrar la secuencia peptídica que codifican en datos de carácter proteómico.

Muchos de los objetivos iniciales no han podido cumplirse, ya que quizá eran demasiado atrevidos dada la naturaleza de los datos y era necesario completar el análisis proteómico de los datos del CPTAC para poder darles respuesta. Como ejemplos de estos objetivos iniciales constaban la predicción de la variabilidad en una secuencia peptídica en células cancerosas, el estudio de otros tipos de alteraciones causadas por mutaciones missense neutrales que afecten a otros aminoácidos además de la leucina y la arginina, o la relación entre la cantidad de mutaciones missense neutrales y el tipo o estado de un cáncer determinado.

Aunque es cierto que los objetivos anteriores no pudieron cumplirse debido a la planificación del trabajo y la falta de conocimientos en el campo de la proteómica, estoy satisfecho de poder haber relacionado datos procedentes de tres ómicas distintas, como la genómica, la transcriptómica y la proteómica, para llevar a cabo un análisis que permita detectar secuencias peptídicas correspondientes a genes seleccionados a través de datos genómicos y transcriptómicos. Además, también he podido confirmar la hipótesis en que los genes con alto contenido GC se expresan en mayor medida que los genes con bajo porcentaje de GC, tal y como se ha demostrado mediante test estadísticos no paramétricos en el apartado de **Discusión** en base a los datos de expresión génica de F. Iorio *et al* (2016) [41]. Adicionalmente, mediante tests estadísticos como el t de Student, se pudo confirmar que no existían diferencias significativas entre los niveles de expresión génica de los genes problema entre células cancerosas y normales, usando los datos de RNA-Seq del GDC. Cabe mencionar que para este mismo dataset de RNA-Seq del GDC se rechazó la hipótesis en que los genes con más contenido GC se expresan más que aquellos con menor porcentaje de GC tanto para células cancerosas como para células sanas.

En resumen, dado que si es posible desearía poder seguir trabajando en este proyecto del Dr. Miquel Àngel Pujana, los resultados y conclusiones que he podido obtener a lo largo de este trabajo, así como la experiencia y conocimientos adquiridos para trabajar con datos de diferentes ómicas, me ayudarán a cumplir los objetivos mencionados a largo plazo.

Glosario

CPTAC: Cancer Institute's Clinical Proteomic Tumor Analysis Consortium
PNNL: Pacific Northwest National Laboratory
VU/VUMC: Universidad de Vanderbilt
TCGA: The Cancer Genome Atlas
MS: Espectrometría de masas
MGF: Mascot Generic Format
NIST: National Institute for Standards and Technology
CRG: Centre for Genomic Regulation
UPF: Universitat Pompeu Fabra
FDR: False Discovery Rate
PSM: Peptide-Spectrum Match
EBI-EMBL: European Bioinformatics Institute
NCBI: National Center for Biotechnology Information
GEO: Gene Expression Omnibus
UniProt: Universal Protein Resource
PDB: Protein Data Bank
HMDB: Human Metabolome Database
KEGG: Kyoto Encyclopedia of Genes and Genomes
SNP: Single Nucleotide Polymorphisms
DNA: Deoxyribonucleic acid
RNA: Ribonucleic acid
PTM: Post-translational modification
ESI: Electrospray ionization
MALDI: Matrix-Assisted Laser Desorption/Ionization
TOF: Time-of-flight
FT: Fourier transform
LC: Liquid chromatography
TOPPAS: The OpenMS Proteomics Pipeline
IDIBELL: Institut d'Investigació Biomèdica de Bellvitge
KNIME: Konstanz Information Miner
GDC: Genomic Data Commons
IDE: Integrated Development Environment
idXML: IdentityXML
NCI: National Cancer Institute
TB: Terabyte
GB: Gigabyte
LFQ: Label-free global proteomic profiling
TMT: Tandem Mass Tags
HUPO: Human Proteome Organization
PCC: Proteome Characterization Centers
CNV: Copy number variation
FPKM-UQ: Fragments Per Kilobase of transcript per Million mapped reads upper quartile
HTSeq: High-throughput sequencing
RNA-Seq: RNA sequencing
CSV: Comma-separated values
COSMIC: Catalogue of Somatic Mutations in Cancer
HapMap: Haplotype Map
TXT: Text
XML: Extensible Markup Language

Bibliografía

[1] M. Altaf-Ul-Amin, F. M. Afendi, S. K. Kiboi, and S. Kanaya, “Systems Biology in the Context of Big Data and Networks,” *Biomed Res. Int.*, vol. 2014, pp. 1–11, 2014.

[2] G. Bell, T. Hey, and A. Szalay, “Computer science. Beyond the data deluge.,” *Science*, vol. 323, no. 5919, pp. 1297–8, Mar. 2009.

[3] F. D. Mast, A. V. Ratushny, and J. D. Aitchison, “Systems cell biology,” *J. Cell Biol.*, vol. 206, no. 6, pp. 695–706, Sep. 2014.

[4] M. Cvijovic *et al.*, “Bridging the gaps in systems biology,” *Mol. Genet. Genomics*, vol. 289, no. 5, pp. 727–734, Oct. 2014.

[5] M. Á. Medina, “Systems biology for molecular life sciences and its impact in biomedicine,” *Cell. Mol. Life Sci.*, vol. 70, no. 6, pp. 1035–1053, Mar. 2013.

[6] J. D. Aitchison and T. Galitski, “Inventories to insights.,” *J. Cell Biol.*, vol. 161, no. 3, pp. 465–9, May 2003.

[7] T. Ideker, T. Galitski, and L. Hood, “A new approach to decoding life: Systems Biology,” *Annu. Rev. Genomics Hum. Genet.*, vol. 2, no. 1, pp. 343–372, Sep. 2001.

[8] P. Nurse, “Life, logic and information,” *Nature*, vol. 454, no. 7203, pp. 424–426, Jul. 2008.

[9] J. A. Alfaro, A. Ignatchenko, V. Ignatchenko, A. Sinha, P. C. Boutros, and T. Kislinger, “Detecting protein variants by mass spectrometry: a comprehensive study in cancer cell-lines.,” *Genome Med.*, vol. 9, no. 1, p. 62, 2017.

[10] L. M. Smith, N. L. Kelleher, and Consortium for Top Down Proteomics, “Proteoform: a single term describing protein complexity.,” *Nat. Methods*, vol. 10, no. 3, pp. 186–7, Mar. 2013.

[11] B. Short, “Cell biologists expand their networks.,” *J. Cell Biol.*, vol. 186, no. 3, pp. 305–11, Aug. 2009.

[12] EMBL-EBI, “What is genomics? | EMBL-EBI Train online,” 2018. [Online]. Available: https://www.ebi.ac.uk/training/online/course/genomics-introduction-ebi-resources/what-genomics. [Accessed: 25-May-2018].

[13] A. Tefferi, “Genomics Basics: DNA Structure, Gene Expression, Cloning, Genetic Mapping, and Molecular Tests,” *Semin. Cardiothorac. Vasc. Anesth.*, vol. 10, no. 4, pp. 282–290, Dec. 2006.

[14] R. Lowe, N. Shirley, M. Bleackley, S. Dolan, and T. Shafee, “Transcriptomics technologies.,” *PLoS Comput. Biol.*, vol. 13, no. 5, p. e1005457, May 2017.

[15] H. F. Willard and G. S. Ginsburg, *Essentials of genomic and personalized medicine*. Academic, 2010.

[16] A. Misra, *Challenges in delivery of therapeutic genomics and proteomics*. Elsevier, 2010.

[17] O. A. Aboud and R. H. Weiss, “New opportunities from the cancer metabolome.,” *Clin. Chem.*, vol. 59, no. 1, pp. 138–46, Jan. 2013.

[18] Y.-Y. Zhao and R.-C. Lin, “UPLC–MSE application in disease biomarker discovery: The discoveries in proteomics to metabolomics,” *Chem. Biol. Interact.*, vol. 215, pp. 7–16, May 2014.

[19] C. Junot, F. Fenaille, B. Colsch, and F. Bécher, “High resolution mass spectrometry based techniques at the crossroads of metabolic pathways,” *Mass Spectrom. Rev.*, vol. 33, no. 6, pp. 471–500, Nov. 2014.

[20] S. Collino, F.-P. J. Martin, and S. Rezzi, “Clinical metabolomics paves the way towards future healthcare strategies,” *Br. J. Clin. Pharmacol.*, vol. 75, no. 3, pp. 619–629, Mar. 2013.

[21] Pearson Education, “Pearson - The Biology Place,” 2018. [Online]. Available: http://www.phschool.com/science/biology_place/glossary/a.html. [Accessed: 26-May-2018].

[22] S. A. MacKenzie and S. Jentoft, *Genomics in aquaculture*. Academic Press, 2016.

[23] Z. Zhang, S. Wu, D. L. Stenoien, and L. Paša-Tolić, “High-Throughput Proteomics,” *Annu. Rev. Anal. Chem.*, vol. 7, no. 1, pp. 427–454, Jun. 2014.

[24] R. Aebersold and M. Mann, “Mass spectrometry-based proteomics,” *Nature*, vol. 422, no. 6928, pp. 198–207, Mar. 2003.

[25] S. Mehmood, T. M. Allison, and C. V. Robinson, “Mass Spectrometry of Protein Complexes: From Origins to Applications,” *Annu. Rev. Phys. Chem.*, vol. 66, no. 1, pp. 453–474, Apr. 2015.

[26] O. D. American Society for Mass Spectrometry., *Journal of the American Society for Mass Spectrometry.*, vol. 12, no. 11. American Society for Mass Spectrometry, 2000.

[27] H. Mirzaei and M. (Scientist) Carrasco, *Modern proteomics : sample preparation, analysis and practical applications*. .

[28] F. E. Ahmed, “Utility of mass spectrometry for proteome ana lysis: part I. Conceptual and experimental approaches,” *Expert Rev. Proteomics*, vol. 5, no. 6, pp. 841–864, Dec. 2008.

[29] F. A. Mellon, “MASS SPECTROMETRY | Principles and Instrumentation,” in *Encyclopedia of Food Sciences and Nutrition*, Elsevier, 2003, pp. 3739–3749.

[30] W. M. . Niessen, “MS–MS and MSn,” in *Encyclopedia of Spectroscopy and Spectrometry*, 2017, pp. 936–941.

[31] M. Gao, H. Zhou, and J. Skolnick, “Insights into Disease-Associated Mutations in the Human Proteome through Protein Structural Analysis.,” *Structure*, vol. 23, no. 7, pp. 1362–

9, Jul. 2015.

[32] G. Corso *et al.*, “BRCA1/2 germline missense mutations,” *Eur. J. Cancer Prev.*, vol. 27, no. 3, p. 1, Mar. 2017.

[33] A. Hijikata, T. Tsuji, M. Shionyu, and T. Shirai, “Decoding disease-causing mechanisms of missense mutations from supramolecular structures,” *Sci. Rep.*, vol. 7, no. 1, p. 8541, Dec. 2017.

[34] S. V Tavtigian and G. Chenevix-Trench, “Growing recognition of the role for rare missense substitutions in breast cancer susceptibility.,” *Biomark. Med.*, vol. 8, no. 4, pp. 589–603, 2014.

[35] D. N. (David N. Cooper and Nature Publishing Group., *Nature encyclopedia of the human genome*. London ;;New York: Nature Pub. Group, 2003.

[36] O. Tomoko, “Synonymous and nonsynonymous substitutions in mammalian genes and the nearly neutral theory,” *J. Mol. Evol.*, vol. 40, no. 1, pp. 56–63, Jan. 1995.

[37] Amino Acids Guide, “Amino Acids - structures, advantages, properties, classifications,” 2018. [Online]. Available: http://www.aminoacidsguide.com/. [Accessed: 03-Jun-2018].

[38] Sigma-Aldrich, “Amino Acids Reference Chart | Sigma-Aldrich,” 2018. [Online]. Available: https://www.sigmaaldrich.com/life-science/metabolomics/learning-center/amino-acid-reference-chart.html. [Accessed: 03-Jun-2018].

[39] Y. Benjamini and T. P. Speed, “Summarizing and correcting the GC content bias in high-throughput sequencing,” *Nucleic Acids Res.*, vol. 40, no. 10, pp. e72–e72, May 2012.

[40] G. Kudla, L. Lipinski, F. Caffin, A. Helwak, and M. Zylicz, “High Guanine and Cytosine Content Increases mRNA Levels in Mammalian Cells,” *PLoS Biol.*, vol. 4, no. 6, p. e180, May 2006.

[41] F. Iorio *et al.*, “A Landscape of Pharmacogenomic Interactions in Cancer.,” *Cell*, vol. 166, no. 3, pp. 740–754, Jul. 2016.

[42] CPTAC, “CPTAC | Office of Cancer Clinical Proteomics Research,” 2018. [Online]. Available: https://proteomics.cancer.gov/programs/cptac. [Accessed: 27-May-2018].

[43] P. A. Rudnick *et al.*, “A Description of the Clinical Proteomic Tumor Analysis Consortium (CPTAC) Common Data Analysis Pipeline,” *J. Proteome Res.*, vol. 15, no. 3, pp. 1023–1032, Mar. 2016.

[44] N. J. Edwards *et al.*, “The CPTAC Data Portal: A Resource for Cancer Proteomics Research,” *J. Proteome Res.*, vol. 14, no. 6, pp. 2707–2713, Jun. 2015.

[45] R. C. Rivers, C. Kinsinger, E. S. Boja, T. Hiltke, M. Mesri, and H. Rodriguez, “Linking cancer genome to proteome: NCI’s investment into proteogenomics,” *Proteomics*, vol. 14, no. 23–24, pp. 2633–2636, Dec. 2014.

[46] Z. Zhang *et al.*, “A survey and evaluation of Web-based tools/databases for variant analysis of TCGA data,” *Brief. Bioinform.*, Mar. 2018.

[47] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, “Review The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge,” *Współczesna Onkol.*, vol. 1A, no. 1A, pp. 68–77, 2015.

[48] H. Lee, J. Palm, S. M. Grimes, and H. P. Ji, “The Cancer Genome Atlas Clinical Explorer: a web and mobile interface for identifying clinical–genomic driver associations,” *Genome Med.*, vol. 7, no. 1, p. 112, Dec. 2015.

[49] D. H. Lundgren, S.-I. Hwang, L. Wu, and D. K. Han, “Role of spectral counting in quantitative proteomics,” *Expert Rev. Proteomics*, vol. 7, no. 1, pp. 39–53, Feb. 2010.

[50] R. D. Smith and D. C. Liebler, “CPTAC - CPTAC Colon Cancer Confirmatory Study,” 2018. [Online]. Available: https://cptac-data-portal.georgetown.edu/cptac/s/S037. [Accessed: 28-May-2018].

[51] U. Christians, S. McCrery, J. Klawitter, and J. Klawitter, “The Role of Proteomics in the Study of Kidney Diseases and in the Development of Diagnostic Tools,” in *Biomarkers of Kidney Disease*, Elsevier, 2011, pp. 101–176.

[52] A. J. R. and S. Weiss, “Epidemiologic and Population Genetic Studies,” in *Clinical and Translational Science*, Elsevier, 2009, pp. 289–299.

[53] W. Zhu, J. W. Smith, and C.-M. Huang, “Mass Spectrometry-Based Label-Free Quantitative Proteomics,” *J. Biomed. Biotechnol.*, vol. 2010, pp. 1–6, 2010.

[54] bioproximity, “Global Proteomic Profiling - Bioproximity,” 2018. [Online]. Available: https://www.bioproximity.com/global-proteomic-analysis-profiling. [Accessed: 28-May-2018].

[55] B. Sun and Q.-Y. He, “Hunting Molecular Targets for Anticancer Reagents by Chemical Proteomics,” in *Novel Approaches and Strategies for Biologics, Vaccines and Cancer Therapies*, Elsevier, 2015, pp. 347–363.

[56] M. Mumby and D. Brekken, “Phosphoproteomics: new insights into cellular signaling.,” *Genome Biol.*, vol. 6, no. 9, p. 230, 2005.

[57] CPTAC, “Common Data Analysis Pipeline (CDAP),” 2018. [Online]. Available: https://cptac-data-portal.georgetown.edu/cptac/aboutData/show?scope=dataLevels#mzML. [Accessed: 28-May-2018].

[58] HUPO-PSI, “HUPO-PSI Working Groups and Outputs | HUPO Proteomics Standards Initiative,” 2018. [Online]. Available: http://www.psidev.info/. [Accessed: 28-May-2018].

[59] National Institutes of Health, “Proteome Characterization Centers - TCGA,” 2018. [Online]. Available:

https://cancergenome.nih.gov/abouttcga/overview/howitworks/proteomecharacterization. [Accessed: 28-May-2018].

[60] H. L. Röst *et al.*, “OpenMS: a flexible open-source software platform for mass spectrometry data analysis,” *Nat. Methods*, vol. 13, no. 9, pp. 741–748, Sep. 2016.

[61] S. Anders, P. T. Pyl, and W. Huber, “HTSeq--a Python framework to work with high-throughput sequencing data,” *Bioinformatics*, vol. 31, no. 2, pp. 166–169, Jan. 2015.

[62] Genomic Data Commons, “HTSeq-FPKM-UQ - GDC Docs,” 2018. [Online]. Available: https://docs.gdc.cancer.gov/Encyclopedia/pages/HTSeq-FPKM-UQ/. [Accessed: 28-May-2018].

[63] The R Foundation, “R: What is R?,” 2018. [Online]. Available: https://www.r-project.org/about.html. [Accessed: 28-May-2018].

[64] RStudio, “RStudio – Take control of your R code,” 2018. [Online]. Available: https://www.rstudio.com/products/RStudio/. [Accessed: 28-May-2018].

[65] tidyverse.org, “A Grammar of Data Manipulation • dplyr,” 2018. [Online]. Available: https://dplyr.tidyverse.org/. [Accessed: 28-May-2018].

[66] BioMart Community, “BioMart,” 2018. [Online]. Available: http://biomart.org/. [Accessed: 28-May-2018].

[67] S. Durinck *et al.*, “BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis,” *Bioinformatics*, vol. 21, no. 16, pp. 3439–3440, Aug. 2005.

[68] J. E. Elias and S. P. Gygi, “Target-Decoy Search Strategy for Mass Spectrometry-Based Proteomics,” in *Methods in molecular biology (Clifton, N.J.)*, vol. 604, 2010, pp. 55–71.

[69] OpenMS, “XTandemAdapter,” 2018. [Online]. Available: http://ftp.mi.fu-berlin.de/pub/OpenMS/release-documentation/html/TOPP_XTandemAdapter.html. [Accessed: 02-Jun-2018].

[70] D. J. Slotta, M. A. McFarland, and S. P. Markey, “MassSieve: panning MS/MS peptide data for proteins.,” *Proteomics*, vol. 10, no. 16, pp. 3035–9, Aug. 2010.

[71] OpenMS, “IDMerger,” 2018. [Online]. Available: http://ftp.mi.fu-berlin.de/pub/OpenMS/release-documentation/html/TOPP_IDMerger.html. [Accessed: 02-Jun-2018].

[72] OpenMS, “FalseDiscoveryRate,” 2018. [Online]. Available: http://ftp.mi.fu-berlin.de/pub/OpenMS/release-documentation/html/TOPP_FalseDiscoveryRate.html. [Accessed: 02-Jun-2018].

[73] OpenMS, “IDFilter,” 2018. [Online]. Available: http://ftp.mi.fu-berlin.de/pub/OpenMS/release-documentation/html/TOPP_IDFilter.html. [Accessed: 02-Jun-2018].

[74] A. W. Kurian *et al.*, “Breast and Ovarian Cancer Penetrance Estimates Derived From Germline Multiple-Gene Sequencing Results in Women,” *JCO Precis. Oncol.*, no. 1, pp. 1–12, Jul. 2017.

[75] National Center for Biotechnology Information, “National Center for Biotechnology Information,” 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/. [Accessed: 02-Jun-2018].

[76] GeneCards, “GeneCards - Human Genes | Gene Database | Gene Search,” 2018. [Online]. Available: https://www.genecards.org/. [Accessed: 02-Jun-2018].

Anexos

Script 1: creación de directorios de trabajo y para las carpetas “data” y “results” donde se almacenaron, respectivamente, los datos y los resultados de los análisis realizados:

```
#Creación de directorios para los datos y los resultados obtenidos,
así como la definición del directorio de trabajo.

setwd("D:/Universidad/Máster Oficial Bioinformática y Bioestadística
UOC-UB/2o Semestre/Treball de Final de Máster/Scripts")
workingDir <- getwd()
system("mkdir data")
system("mkdir results")
dataDir <- file.path(workingDir, "data")
resultsDir <- file.path(workingDir, "results")
```

Script 2: lectura de los datos “summary” de los estudios PNNL y VU del ensayo del CPTAC y selección de los genes analizados en función del parámetro Spectral.Counts (el total de espectros identificados para una misma proteína).

```
#Datos "summary" PNNL:

data_pnnl_summary <- read.table(file = 'D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/data/CPTAC_COprospective_PNNL_Proteome_CDAP_Protein_Rep
ort_r1/CPTAC2_Colon_Prospective_Collection_PNNL_Proteome_summary.tsv',
sep = '\t', header = TRUE)

#Datos "summary" VU:

data_vu_summary <- read.table(file = 'D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/data/CPTAC_COprospective_VU_Proteome_CDAP_Protein_Repor
t_r1/CPTAC2_Prospective_Colon_VU_Proteome_summary.tsv', sep = '\t',
header = TRUE)

str(data_pnnl_summary)
str(data_vu_summary)

#Uso del paquete "dplyr":

if(!(require(dplyr))) install.packages("tidyverse")

#Selección de las características relevantes y filtraje de los genes
según su valor Spectral.Counts:

library(dplyr)
data_pnnl <- select(data_pnnl_summary,
                    Gene,
                    Spectral.Counts,
                    Distinct.Peptides,
                    Unshared.Peptides,
                    NCBIGeneID,
```

```
                    Authority,
                    Description,
                    Organism,
                    Chromosome,
                    Locus,
                    Proteins,
                    Assays)
data_pnnl_sel <- filter(data_pnnl, Spectral.Counts >= 100)
data_vu <- select(data_vu_summary,
                    Gene,
                    Spectral.Counts,
                    Distinct.Peptides,
                    Unshared.Peptides,
                    NCBIGeneID,
                    Authority,
                    Description,
                    Organism,
                    Chromosome,
                    Locus,
                    Proteins,
                    Assays)
data_vu_sel <- filter(data_vu, Spectral.Counts >= 100)
total_data <- merge(data_pnnl_sel, data_vu_sel, by = "Gene", suffixes
= c(".PNNL", ".VU"))
write.csv(
  total_data,
  "D:/Universidad//Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenestotal.csv")
total_genes <- as.vector(total_data$Gene)
write.csv(
  total_genes,
  "D:/Universidad//Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenes.csv")
```

Script 3: extracción de datos genómicos del BioMart (Ensembl) de los genes seleccionados en el script anterior. Selección de los 10 genes problema con alto contenido GC y los 10 genes problema con bajo porcentaje GC. Análisis de los datos de expresión génica de los genes problema en función del dataset RNA-Seq del GDC para células cancerosas y para células sanas. Test estadísticos para determinar la normalidad, homocedasticidad y la significancia de los valores de expresión génica de los genes seleccionados.

```
if(!require(biomaRt, dplyr, data.table)){
  source("http://bioconductor.org/biocLite.R")
  biocLite()
  biocLite("biomaRt")
  require(biomaRt)
  require(dplyr)
  require(data.table)
}

#Extracción de los datos de BioMart (Ensembl) de los genes problema:
```

```
mapHgncSymbols <- function(ensembl, symbols){

  mapping <- getBM(attributes = c("entrezgene", "hgnc_symbol",
"ensembl_gene_id",
                                  "gene_biotype",
"percentage_gene_gc_content"),
                   filters = "hgnc symbol" , values = symbols,
                   mart = ensembl)
  return(mapping)
}

ensembl = useMart('ensembl')
ensembl = useDataset("hsapiens_gene_ensembl", mart=ensembl)

symbols = as.data.frame(
  read.csv(
    "D:/Universidad/Máster Oficial Bioinformática y Bioestadística
UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenes.csv",
    row.names = 1,
    col.names = c("", "SYMBOL"),
    stringsAsFactors = FALSE
    )
  )

head(symbols$SYMBOL) #para comprobar si hemos importado correctamente

genes = mapHgncSymbols(ensembl, symbols$SYMBOL)
data_genes = as.data.frame(genes)
write.csv(
  data_genes,
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de Máster/Scripts/results/genesGC.csv"
  )

#Leer datos y mezclar:

total_data_final <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/listagenestotal.csv",
  row.names = 1
  )

genes_gc <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genesGC.csv",
  row.names = 1
)

colnames(genes_gc)[2] <- "Gene" #Cambiamos el nombre de la columna
hgnc_symbol por Gene para poderla relacionar
genes_gc_score <- merge(total_data_final, genes_gc, by = "Gene") #Aquí
tendríamos combinado todo, GC y datos espectrométricos
genes_gc_score <- arrange(
  genes_gc_score,
  genes_gc_score$percentage_gene_gc_content
  )
```

```
#Mediante dyplr, cojemos 10 genes con alto GC y alto Spectral.Counts y
10 genes con bajo GC y alto Spectral.Count:

highgc <- filter(genes_gc_score,
                 genes_gc_score$Spectral.Counts.PNNL > 7500,
                 genes_gc_score$percentage_gene_gc_content >= 60
                 )
write.csv(highgc, "D:/Universidad/Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/high_gc_genes.csv")
lowgc <- filter(genes_gc_score,
                genes_gc_score$Spectral.Counts.PNNL > 6600,
                genes_gc_score$percentage_gene_gc_content <= 40
                )
write.csv(lowgc, "D:/Universidad/Máster Oficial Bioinformática y
Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/low_gc_genes.csv")

#Cargamos los datasets de las bases de datos de lineas celulares:

loadRData = function(path){
  variable = load(path)
  return(get(variable))
}

rnaseqcancer <- loadRData("D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/rnaSeq.RData")
rnaseqnormal <- loadRData("D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/rnaSeqNormals.RData")

#Hacemos una interejección entre los 10 genes seleccionados con alto y
bajo contenido GC (por separado) con los datos RNA-Seq para células
cancerosas y normales:
#Mezclamos en función de la variable "ensembl_gene_id".

genes_high_gc_cancer <- merge(highgc, rnaseqcancer, by =
"ensembl_gene_id")
genes_high_gc_normal <- merge(highgc, rnaseqnormal, by =
"ensembl_gene_id")
genes_low_gc_cancer <- merge(lowgc, rnaseqcancer, by =
"ensembl_gene_id")
genes_low_gc_normal <- merge(lowgc, rnaseqnormal, by =
"ensembl_gene_id")


#Media de los valores de expresión de los genes seleccionados en
células tumorales:

genes_high_gc_cancer_mean <- transform(
  genes_high_gc_cancer,
  TCGA = rowMeans(genes_high_gc_cancer[,30:1131],na.rm = TRUE)
  )
genes_high_gc_cancer_mean <- genes_high_gc_cancer_mean[,-c(30:1131)]

genes_low_gc_cancer_mean <- transform(
  genes_low_gc_cancer,
  TCGA = rowMeans(genes_low_gc_cancer[,30:1131],na.rm = TRUE)
  )
genes_low_gc_cancer_mean <- genes_low_gc_cancer_mean[,-c(30:1131)]
```

```
write.csv(genes_high_gc_cancer_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_high_gc_cancer_mean.csv")
write.csv(genes_low_gc_cancer_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_low_gc_cancer_mean.csv")


#Media de los valores de expresión de los genes seleccionados en
células normales:

genes_high_gc_normal_mean <- transform(
  genes_high_gc_normal,
  TCGA = rowMeans(genes_high_gc_normal[,30:142],na.rm = TRUE)
)
genes_high_gc_normal_mean <- genes_high_gc_normal_mean[,-c(30:142)]

genes_low_gc_normal_mean <- transform(
  genes_low_gc_normal,
  TCGA = rowMeans(genes_low_gc_normal[,30:142],na.rm = TRUE)
)
genes_low_gc_normal_mean <- genes_low_gc_normal_mean[,-c(30:142)]

write.csv(genes_high_gc_normal_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_high_gc_normal_mean.csv")
write.csv(genes_low_gc_normal_mean, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/genes_low_gc_normal_mean.csv")

#Tests estadísticos

#Test de Shapiro-Wilk (normalidad):

shapiro.test(genes_high_gc_cancer_mean$TCGA)
shapiro.test(genes_high_gc_normal_mean$TCGA)
shapiro.test(genes_low_gc_cancer_mean$TCGA)
shapiro.test(genes_low_gc_normal_mean$TCGA)

#Test de F de Fisher (homocedasticidad):

var.test(genes_high_gc_cancer_mean$TCGA,
genes_high_gc_normal_mean$TCGA)
var.test(genes_low_gc_cancer_mean$TCGA, genes_low_gc_normal_mean$TCGA)

#Test t de Student:

ttest_highGC <- t.test(genes_high_gc_cancer_mean$TCGA,
genes_high_gc_normal_mean$TCGA, var.equal = TRUE)
ttest_highGC

ttest_lowGC <- t.test(genes_low_gc_cancer_mean$TCGA,
genes_low_gc_normal_mean$TCGA, var.equal = TRUE)
ttest_lowGC

test_cancer <- t.test(genes_high_gc_cancer_mean$TCGA,
genes_low_gc_cancer_mean$TCGA, var.equal = TRUE)
test_cancer
```

```
test_normal <- t.test(genes_high_gc_normal_mean$TCGA,
genes_low_gc_normal_mean$TCGA, var.equal = TRUE)
test_normal
```

Script 4: lectura de los datos de líneas tumorales y datos de expresión génica del estudio de F. Iorio *et al* (2016) [41]. Análisis de los niveles de expresión de los 10 genes problema con alto %GC y los 10 genes con bajo %GC. Test estadísticos para determinar la normalidad, homocedasticidad y significancia de los valores de expresión génica de los genes problema.

```
if(!require(dplyr, xlsx)){
  library(dplyr)
  library(xlsx)
}

loadRData = function(path){
  variable = load(path)
  return(get(variable))
}

#Lectura de los datos de lineas tumorales y de los datos de expresión
génica:

samples <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/data/mm5_samples_by_cancer_type.csv",
  header = T,
  sep = ";"
  )

str(samples)

samples_coread <- samples %>% filter(samples$Cancer.Type ==
"COAD/READ")
expression_data <- loadRData(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/data/expressionData.Rdata"
  )

#Procesamiento de los datos de expresión génica para su análisis:

expression_data <- as.matrix(expression_data)
expression_data_rows <- as.matrix(rownames(expression_data))
colnames(expression_data_rows) <- "Identifier"
expression_data_combo <- cbind(expression_data, expression_data_rows)

#Mezclamos ambos datasets con "merge" y tomando como referencia el
identificador.

expression_cancer <- merge(samples_coread, expression_data_combo, by =
"Identifier")

#Genes con alto contenido GC:
```

```
high_gc_genes <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/high_gc_genes.csv",
  header = T
)

expression data filter highgc <- colnames(expression cancer) %in% c(

"GAPDH","FLNC","ACTG1","HIST2H2AA3","FLNA","HIST1H3D","ACTA1","COL6A1"
,"H2AFX","HBA1")
expression_data_highgc <-
expression_cancer[expression_data_filter_highgc]
expression_data_highgc <- apply(expression_data_highgc, 2, as.numeric)
expression_data_highgc <-
as.data.frame(colMeans(expression_data_highgc))
colnames(expression_data_highgc) <- "Promedios"
write.csv(expression_data_highgc, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/expression_data_highgc.csv")

#Genes con bajo contenido GC:

low_gc_genes <- read.csv(
  "D:/Universidad/Máster Oficial Bioinformática y Bioestadística UOC-
UB/2o Semestre/Treball de Final de
Máster/Scripts/results/low_gc_genes.csv",
  header = T
)
lowgc_names <- as.data.frame(t(data.frame("Genes" =
low_gc_genes$Gene))) #extraemos los nombres de los genes

expression_data_filter_lowgc <- colnames(expression_cancer) %in% c(

"ALB","FGB","COL12A1","A2M","COL1A2","HBB","HBD","FBN1","FN1","HSP90B1
")
expression_data_lowgc <-
expression_cancer[expression_data_filter_lowgc]
expression_data_lowgc <- apply(expression_data_lowgc, 2, as.numeric)
expression_data_lowgc <-
as.data.frame(colMeans(expression_data_lowgc))
colnames(expression_data_lowgc) <- "Promedios"
write.csv(expression_data_lowgc, "D:/Universidad/Máster Oficial
Bioinformática y Bioestadística UOC-UB/2o Semestre/Treball de Final de
Máster/Scripts/results/expression_data_lowgc.csv")

#Tests estadísticos

#Test de Shapiro-Wilk (normalidad):

shapiro.test(expression_data_highgc$Promedios)
shapiro.test(expression_data_lowgc$Promedios)

#Test de Levene:

grupomuestras <- c(expression_data_highgc$Promedios,
expression_data_lowgc$Promedios)
grupo <- as.factor(c(rep(1, length(expression_data_highgc$Promedios)),
rep(2, length(expression_data_lowgc$Promedios))))
library(Rcmdr)
leveneTest(grupomuestras, grupo)
```

```
#Test de Mann-Whitney U:

#wilcox.test(y,x)
wilcox.test(expression_data_lowgc$Promedios,
expression_data_highgc$Promedios)

#Promedio de los valores de expresión totales de los genes problema:

mean(expression_data_highgc$Promedios)
mean(expression_data_lowgc$Promedios)
```