UNIVERSITAT ROVIRA i VIRGILI

Análisis de expresión diferencial en células PBMC de pacientes sanos vs COVID-19

Marcos Esteve Hernández

TRABAJO FINAL DE GRADO BIOTECNOLOGÍA

Tutora académica: Dra. María Teresa Blay Olivé

Fecha de convocatoria: Junio 2021

Yo, Marcos Esteve Hernández, con DNI 46089542A, soy conocedor de la guía de prevención del plagio en la URV *Prevenció, detecció i tractament del plagi en la docència: guía per a estudiants* (aprobada en julio 2017) (http://www.urv.cat/ca/vidacampus/serveis/crai/que-us-oferim/formacio-competencies-nuclears/plagi/) y afirmo que este TFG no constituye ninguna de las conductas consideradas como plagio por la URV.

Tarragona, 3 de Junio de 2021

ÍNDICE

Abstract

Este trabajo comienza con un escueto recorrido por las técnicas y tecnologías de secuenciación que han surgido de unos años hacia ahora, hasta llegar a la técnica en la que se basa este análisis y que posibilita que se pueda realizar, la *single-cell RNA sequencing* (*scRNA-seq*). Se desarrolla un análisis de expresión diferencial de células mononucleares periféricas de la sangre (PBMC) que han sido secuenciadas a través de la scRNA-seq, ya que estas células son las que más participan en la respuesta inmune y nos dan una visión de los mecanismos tanto del individuo como del virus durante el proceso de infección. Se han escogido como punto de partida del análisis los resultados de la secuenciación de estas células en 7 pacientes con la enfermedad del coronavirus 2019 (COVID-19) y 6 pacientes sanos como control. El motivo de esta elección es debido al surgimiento de la pandemia provocada por esta enfermedad y la necesidad de ampliar nuestro conocimiento sobre el virus y los cambios que provoca en el individuo enfermo. Se realiza un recorrido exhaustivo por cada paso que compone el análisis, clarificando el motivo de su ejecución y exponiendo su resultado, con el objetivo de ofrecer una visión clara y que sirva como guía de aprendizaje para quienes quieran realizar este tipo de análisis. Finalmente, los resultados obtenidos son expuestos y discutidos en referencia a la función de los genes con diferencia de expresión en ambos grupos. Estos resultados exponen mecanismos de acción tanto de las células infectadas como del virus, y abren la puerta a futuras investigaciones para conocer más en detalle la fisiopatología de la enfermedad.

Palabras clave: Single Cell RNA-seq Analysis, PBMC, COVID-19, Differential Gene Expression, Bioinformatics.

Introducción

A principios del siglo XXI surgieron, gracias a las nuevas tecnologías y al avance en campos como la informática y la biología, las llamadas técnicas de secuenciación de nueva generación (*Next Generation Sequencing*, NGS). Estas técnicas las componen un conjunto de tecnologías diseñadas para secuenciar gran cantidad de fragmentos de DNA o RNA de forma masiva y en paralelo (*Rubio et al*, 2020), lo que nos permite obtener resultados en menor tiempo y con menor coste por base. Es por ello por lo que estas técnicas también son llamadas *High-throughput Sequencing* (HTS), secuenciación de alto rendimiento, debido a la cantidad masiva de datos que podemos secuenciar en muy poco tiempo en comparación con las técnicas clásicas de secuenciación, también llamadas de primera generación (FGS).

A pesar de que cada método utiliza sus propias técnicas o está basado en principios diferentes, las tecnologías NGS están basadas en un conjunto de métodos para preparar 'templates' o plantillas de DNA, llevar a cabo lecturas de forma paralela de millones de fragmentos de DNA, técnicas de captura de imagen en tiempo real, alineamiento de secuencias, ensamblaje de secuencias y detección de variantes (*Ari et al*, 2016).

El principal objetivo de estas tecnologías es el de estudiar las variaciones genéticas asociadas con enfermedades u otros fenómenos biológicos. Las técnicas NGS han permitido que se creen nuevas áreas de investigación biológica como la dinámica transcriptómica, la estructura genómica o la variación genómica (*Soon et al*, 2013) y les han dado la oportunidad a los investigadores de estudiar sistemáticamente las interacciones entre estructura y función de los sistemas biológicos de un modo que no había sido posible anteriormente (*Ari et al*, 2016).

Dentro de todas las aplicaciones y los campos que permiten estudiar las técnicas NGS, en este trabajo se va a hacer especial énfasis en el campo de la transcriptómica, el estudio del conjunto de RNA de una célula, tejido u órgano. Cada una de las distintas moléculas de RNA (mRNA, rRNA, tRNA, miRNA, etc.) juega un papel importante en la respuesta fisiológica, por lo que su estudio es vital para entender el genoma funcional. La técnica más utilizada en la actualidad para secuenciar RNA es la *RNA-sequencing*, debido a que, en contraposición con la secuenciación por Sanger, esta provee mayor cobertura y resolución, además de una reducción de tiempo considerable (*Kukurba et al*, 2015). Por medio de esta técnica es posible llevar a cabo una cuantificación precisa de la expresión de los genes; pero además, los datos que recogemos de la *RNA-sequencing* también facilitan el descubrimiento de nuevos transcritos, la identificación

de '*spliced genes*' alternativos y la detección de expresiones alelo-específicas (*Kukurba et al, 2015*). Otras ventajas que ofrece esta técnica respecto a las basadas en *microarrays* son: no es necesario un conocimiento previo del genoma para llevarla a cabo, es más precisa a la hora de evaluar el 'fold-change' (cuánto cambia una cantidad entre una medición original y una posterior) tanto en genes que presentan mucha expresión como en aquellos que presentan poca, y la capacidad de obviar fácilmente las lecturas que se consideren ambiguas, lo que resulta en una reducción del ruido a la hora de hacer el análisis de los datos (*Anamika et al*, 2016).

Las técnicas de *RNA-sequencing* analizan la expresión de RNAs en grandes poblaciones de células; sin embargo, en aquellas poblaciones mixtas, en las que están presentes gran variedad de tipos celulares, estos análisis pueden tapar o esconder grandes diferencias entre las células de un mismo tipo celular (*Shapiro et al*, 2013).

La técnica *single-cell RNA sequencing* (scRNA-seq) proporciona perfiles de expresión de células individuales y está considerada como un 'gold standard' (se llama 'gold-standard' a una prueba que supone la mejor opción para llevar a cabo una tarea bajo unas condiciones razonables) para definir estados celulares y fenotipos (*Tanay & Regev*, 2017). Esta técnica permite identificar y descubrir nuevos patrones de expresión mediante el análisis del agrupamiento de genes en clústeres. Es por esto que, cuando se quiere investigar sobre la transcriptómica de uno o varios tipos celulares en concreto, se recurre a esta técnica que proporciona unos resultados más profundos sobre el fenotipo de células concretas y da la posibilidad de visualizar posibles características que con otras técnicas podrían quedar opacadas.

Con el objetivo de ilustrar y explicar de una manera más clara en qué consiste la metodología de la scRNA-seq, se ha incluido la *Figura 1*. Esta imagen muestra el esquema de trabajo cuando se lleva a cabo una secuenciación por medio de la técnica scRNA-seq. La secuenciación parte del tejido que cuenta con las células el transcriptoma de las cuales se quiere estudiar. Se extrae parte del tejido celular y se procede a aislar las células. Cuando se tienen las células aisladas -generalmente cada célula en un pocillo individual- se extrae el RNA de estas, comúnmente utilizando algún solvente orgánico como el fenol y el cloroformo o mediante ultracentrifugación. A partir de este RNA y mediante la retrotranscriptasa o transcriptasa inversa se obtiene la cadena de DNA complementaria al RNA (cDNA). El cDNA es entonces amplificado mediante PCR cuantitativa (qPCR) y secuenciado, obteniendo una librería. Con estos datos, se elabora el perfil de expresión. El último paso que se puede observar en la *Figura 1*, la clusterización y posterior identificación de los tipos celulares, correspondería

a la parte del análisis de los datos obtenidos por la secuenciación, de lo cual se hablará más en profundidad en el apartado de metodología de este trabajo. Es importante destacar que, a pesar de que esta es la secuencia de pasos que se lleva a cabo en la secuenciación, los métodos que se utilizan en cada paso difieren según la metodología llevada a cabo y según la compañía que la realice, habiendo diferentes estrategias para la síntesis y amplificación del cDNA, la transcripción inversa o el aislamiento celular.

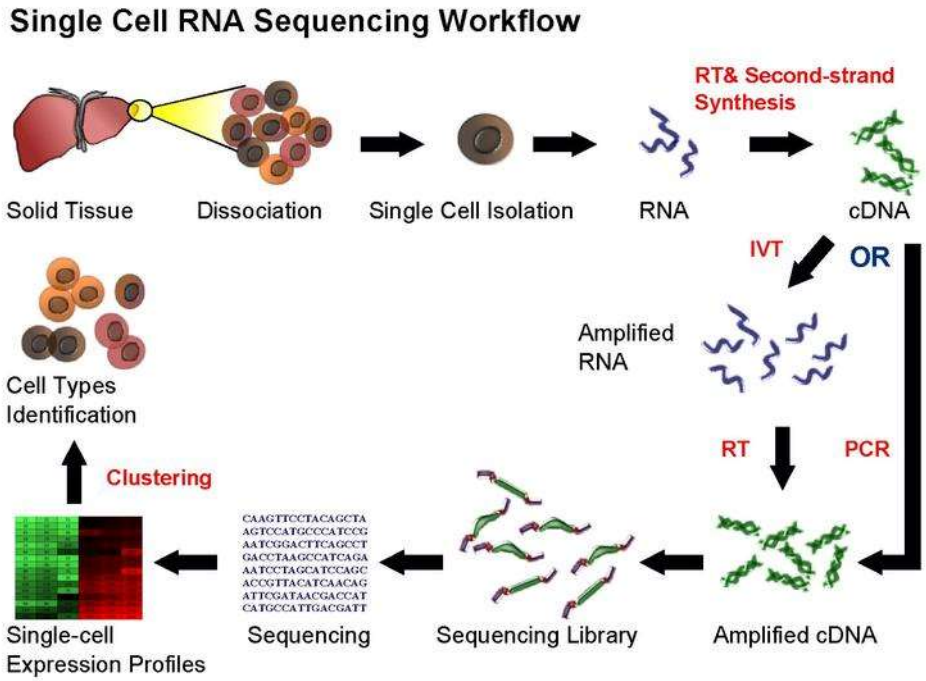

Figura 1. Diagrama del flujo de trabajo en una single-cell RNA sequencing.

Los datos que se obtienen a partir de la secuenciación de scRNA, estos perfiles de expresión, suelen presentarse como matrices de expresión. Estas matrices están compuestas de la siguiente manera: cada una de las filas representa un gen que ha sido secuenciado, mientras que cada una de las columnas representa una célula (esto es por convención aunque podría darse el caso inverso). Por lo tanto, cada entrada de la matriz, es decir, cada intersección entre una fila y una columna representa el nivel de expresión de un gen en particular en una célula concreta. Las unidades de estos valores dependen del protocolo o tecnología que se haya utilizado para llevar a cabo la secuenciación y la estrategia de normalización aplicada. Por lo tanto, el resultado de esta secuenciación es una gran colección de lecturas de cDNA que se agrupan en forma de matriz para facilitar su identificación y posterior análisis. Esta parte es importante ya que esta matriz es la base de donde parte el análisis de expresión diferencial que se lleva a cabo en este trabajo, detallado en la sección de metodología.

Las aplicaciones de la secuenciación de scRNA y el análisis de los datos que ofrece esta se han extendido a lo largo de los años y abarcan multitud de campos de la biología y la medicina. Se han publicado gran cantidad de artículos en los que la transcriptómica de las *single-cell* constituye la clave y el principal camino por el que realizar avances científicos en disciplinas como la neurología (*Raj et al*, 2018), la oncología (*Levitin et al*, 2018), la inmunología (*Neu et al*, 2017) o el estudio de enfermedades infecciosas (*Bossel Ben-Moshe* et al, 2019).

Es en este último campo, en el estudio de enfermedades infecciosas, donde se centraría el trabajo que realizo. En diciembre de 2019 se notificó en Wuhan, China un brote de una enfermedad de causa desconocida, que días más tarde, ya en enero de 2020, se reportó que era causada por una nueva variante de coronavirus que se llamó SARS-CoV-2. Esta enfermedad, denominada *coronavirus disease 2019* (COVID-19), dio lugar a una situación de pandemia mundial que en el momento de escribir estas líneas, en 2021, todavía continúa. Durante todo este tiempo, se ha estado investigando sobre esta nueva enfermedad con el objetivo de ampliar nuestro conocimiento sobre ella y encontrar la mejor forma de combatirla. En esta tarea, la transcriptómica de las *single-cell* ha tenido un papel muy importante, ayudando a conocer las características de la respuesta inmune de los pacientes con COVID-19 con el objetivo de comprender la patogenia de la enfermedad y proporcionar información para la elaboración de estrategias terapéuticas efectivas (*Ren et al*, 2021).

Objetivos

Este trabajo lo he desarrollado teniendo presente dos objetivos principales: el primero, obtener un conocimiento más profundo sobre la enfermedad del COVID-19 y los cambios que genera en el individuo, centrándome en las células que componen principalmente la respuesta inmune y ofrecer a su vez información sobre sus mecanismos de acción para que sirva en futuras investigaciones y tratamientos específicos. El segundo objetivo es más tecnológico, ya que busco comprender en detalle cada proceso del análisis de expresión diferencial, el motivo de cada paso, sus entradas y el resultado que generan, para especializarme y poder utilizarlo de cara al ámbito laboral que me espera en terminar el grado, a la vez que ofrezco una explicación detallada para que sirva como beneficio a cualquiera que quiera adentrarse en este campo o ampliar su conocimiento sobre él.

Metodología

Búsqueda de datos iniciales

El análisis que realizo en este trabajo parte de datos obtenidos a partir de otro estudio en el que se ha llevado a cabo una secuenciación de células PBMC (*Peripheral Blood Mononuclear Cell*) mediante el método *single-cell RNA sequencing*, ya explicado en la introducción. Para encontrar un estudio que cumpla con estas condiciones, realicé una búsqueda en la base de datos PubMed (https://pubmed.ncbi.nlm.nih.gov/), la cual está vinculada con la base de datos NCBI (National Center for Biotechnology Information) (https://www.ncbi.nlm.nih.gov/). Situándome en la página de inicio de PubMed, clico en el botón *Advanced* para realizar una búsqueda avanzada. En la página que se muestra a continuación, introduje los términos MeSH "Single-Cell Analysis" y "SARS-CoV-2", ya que son los principales términos que hacen referencia al objetivo de mi búsqueda. Sin embargo, esta búsqueda arrojó resultados de artículos en los que se había secuenciado todo tipo de tejidos. Como a mí solo me interesan aquellos que secuencien células mononucleares periféricas de la sangre, introduje en mi búsqueda el término "Blood", también como MeSH Term. En esta ocasión, el primero de los 21 resultados que arrojó mi búsqueda fue el artículo "A single-cell atlas of the peripheral immune response in patients with severe COVID-19" (Wilk et al., 2020). Este artículo está elaborado por la universidad de Stanford y fue publicado en la revista Nature. Este artículo tiene como objetivo ampliar lo que se conoce sobre la fisiopatología de la COVID-19 a través de la caracterización de la actividad periférica del sistema inmune.

He seleccionado este artículo como base de mi trabajo principalmente porque proporciona los datos extraídos de la *scRNA-sequencing*, las matrices con el recuento de las lecturas. Estos datos se pueden descargar desde la base de datos de GEO (Gene Expression Omnibus) (https://www.ncbi.nlm.nih.gov/geo/). Para acceder a estos datos, en la página de Pubmed correspondiente al artículo de (Wilk et al, 2020) se encuentra el enlace a los datos en la base de datos de GEO; o bien se puede realizar una búsqueda en la página de GEO del título del artículo. La información que encontramos sobre el artículo nos indica que los datos corresponden a 7 pacientes con COVID-19 y 6 pacientes sanos que actúan como control. Esta información me permite identificar en qué condición se encuentra cada muestra y así poder realizar el análisis comparando los datos entre distintas condiciones, algo fundamental en el análisis. Estos datos constituyen el punto de partida de mi trabajo, por lo que este artículo es idóneo para comenzar a desarrollar mi proyecto sobre él.

Análisis de expresión diferencial

El análisis que realizo en este trabajo está basado principalmente en una librería (conjunto de funciones en informática) desarrollada por el grupo SatijaLab (Seurat, https://satijalab.org/seurat/) en el lenguaje de programación R. En la página web desarrollada por ese grupo, proporcionan un repositorio de funciones estadísticas mediante el uso de las cuales se lleva a cabo este análisis, incluyendo funciones desde la normalización y escalado de los datos hasta herramientas más concretas para la agrupación de las células en clústeres. Es por ello por lo que he decidido realizar este trabajo basándome en las guías para utilizar sus funciones que proporcionan en su página web. Cabe destacar que, pese a la existencia de estas guías, solo son una pequeña referencia a cómo se deben usar, los parámetros que más comúnmente se utilizan, pero la correcta ejecución de estas dependerá de los datos que quieras analizar y del enfoque concreto que quieras dar al análisis. Es por ello por lo que mi trabajo ha consistido en la construcción de un script o programa, que se traduce como un conjunto de funciones ordenadas de una forma concreta para llevar a cabo una tarea específica, utilizando tanto las funciones proporcionadas por Seurat como mi propio conocimiento y experiencia en programación para que el resultado del análisis sea correcto y coherente.

El entorno de programación en el que llevo a cabo el análisis es RStudio (https://www.rstudio.com/), el cual incorpora el lenguaje de programación R y tiene acceso a las librerías del grupo Seurat y a otras que serán necesarias. El lenguaje de programación R es software libre, es decir, no es necesario ningún tipo de licencia ya que es gratuito.

Obtención de los datos

El primer paso en este análisis no puede ser otro que el de cargar los datos del estudio escogido en el entorno de trabajo, es decir, en RStudio. Los datos proporcionados en la página de GEO correspondiente a este estudio (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE150728) son las matrices de expresión resultado de la secuenciación y están separadas por pacientes, es decir, hay 7 matrices pertenecientes a 7 pacientes con COVID-19 y 6 matrices que pertenecen a 6 pacientes sanos usados como control.

Este paso es dependiente del formato en el que se presentan los datos del estudio, ya que, pese a que generalmente los resultados de la secuenciación por scRNA-seq son presentados como matrices de expresión, existe variabilidad en el formato que se guardan informáticamente estos datos. Durante mi estancia en prácticas, en la cual

realicé análisis muy similares al que realizo en este trabajo, me encontré con que los datos unas veces se presentan en formato MTX (*Microsoft Visual Studio Manual Test Text Format*), otras en formato H5 (Formato de Datos Jerárquicos v5), y otras, como es el caso del trabajo, en formato RDS. Al presentarse en formato RDS, existe una función en R que se encarga de leer este tipo de archivos y cargarlos en el entorno de trabajo, llamada *readRDS*. A esta función se le debe especificar el archivo que debe leer, por lo que se llamará a esta función una vez para cada archivo de datos (7 Covid-19 y 6 controles sanos, por lo tanto 13 veces) haciendo referencia al archivo correspondiente.

Creación de la estructura de datos

Una vez tenemos los datos en el entorno de trabajo, se debe crear un objeto Seurat con ellos. Este objeto es una estructura de datos proporcionada por Seurat que nos permite tener los datos de una forma concreta para poder aplicar las funciones estadísticas y analíticas que forman parte del análisis de una manera más cómoda. Por lo tanto, usaremos la función *CreateSeuratObject* con los datos que acabamos de cargar para crear un objeto Seurat para cada caso, y a continuación integraremos todos los datos en un único objeto mediante la función *merge*, la cual nos permite combinar objetos o estructuras de datos y etiquetar a cada conjunto de datos con un identificador de manera que posteriormente podamos saber qué genes pertenecen al paciente número 1 o al paciente número 4.

Control de calidad

Una vez tenemos todos los datos integrados, es momento de aplicar una serie de filtros de calidad, es decir, seleccionar únicamente aquellas células con la que vamos a continuar el análisis y descartar aquellas que o bien no sean relevantes o bien nos puedan dar lugar a resultados confusos o que no reflejen del todo la realidad.

Ejemplificando esto, el primer filtro que vamos a aplicar a nuestros datos es descartar aquellas células que presenten un porcentaje elevado de RNA mitocondrial. La presencia en abundancia de RNA mitocondrial en una célula normalmente es un indicativo de que la célula está muriendo, por lo que no queremos utilizar estas células en nuestro análisis ya que podría darnos una idea errónea de lo que está ocurriendo y no podríamos estar seguros de que las lecturas de expresión que obtengamos de ella son las mismas que las de una célula en la plenitud de su vida. Para aplicar este filtro, primero generamos una columna en nuestro objeto Seurat que refleje la cantidad de RNA mitocondrial presente en cada célula mediante la función *PercentageFeatureSet* utilizando como patrón identificativo de la función "^MT-". Este patrón indica a la función que deberá contar aquellos genes cuyo identificativo comience (^) por las letras MT

seguidas de un guión (-). Así, esta columna nos arrojará el número de genes que cumplen este patrón respecto al total de genes de cada célula, es decir, el porcentaje de RNA mitoncodrial.

Podemos visualizar algunas de las características que nos permiten hacer estos controles de calidad, como son el “nFeature”, que es el número de genes detectados en cada célula, el “nCount”, que es el número de moléculas detectadas en cada célula, o el porcentaje de RNA mitocondrial, que ya se ha comentado. Un número bajo de “nFeature” indica que la célula podría estar muerta o muriendo, o que el pocillo se encuentre vacío. Por otro lado, un número elevado de “nFeature” o “nCount” podría indicar que el pocillo es más bien un doblet o un triplet, es decir, que no se realizó bien el aislamiento celular y más de una célula cayó en el mismo pocillo. Es por todo esto que se debe tener en cuenta estos valores y visualizarlos nos puede dar una idea de cómo son o cómo están nuestros datos. Utilizamos la función *VlnPlot* para generar el *Gráfico 1*.

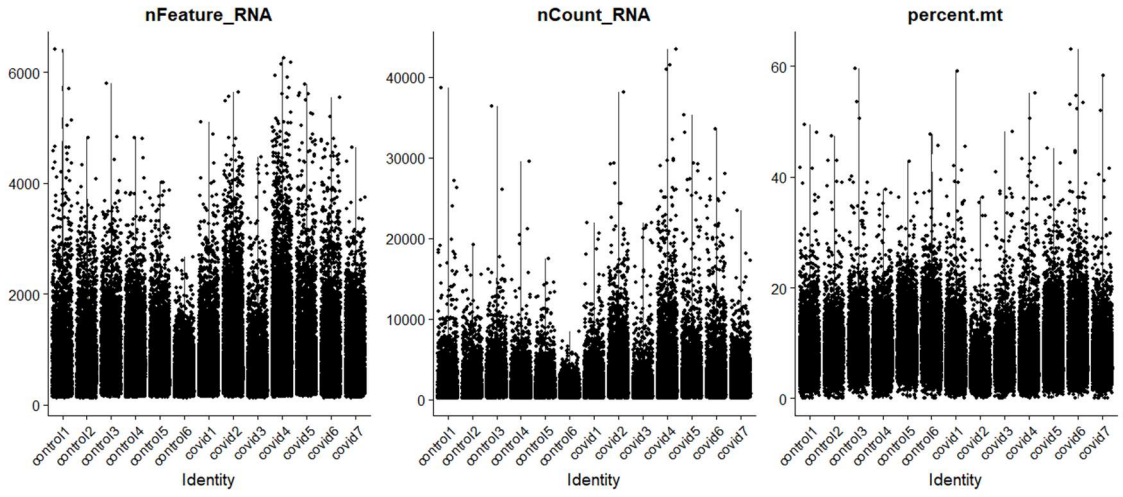

Gráfico 1. Cantidad de genes, moléculas y porcentaje de RNA mitocondrial detectada en cada muestra

En el Gráfico 1 podemos apreciar los valores de las variables ‘nFeature’, ‘nCount’ y ‘percent.mt’ en nuestro conjunto de datos, respectivamente. Cada columna de cada gráfico representa una muestra, es decir, tenemos 13 columnas que se corresponden con los 6 pacientes control y las 7 muestras de pacientes con Covid-19. Fijándonos en el gráfico correspondiente al porcentaje de RNA mitocondrial, podemos observar cómo hay una gran cantidad de células que presentan un elevado porcentaje de este; estas céulas no nos interesan desde el punto de vista del análisis por lo que ya se ha comentado con anterioridad, por lo que serán descartadas.

Ahora que tenemos una idea de la calidad de nuestras muestras, podemos subdividir nuestro conjunto de datos, restringiendo las características que queremos que tengan

los datos seleccionados con los que continuaremos el análisis. En este caso, se ha optado por aplicar la restricción de que el campo “nFeature” se encuentre entre los valores de 200 y 2000, y que el porcentaje de RNA mitocondrial de cada célula no supere el 5%. Estos valores están basados en lo que recomienda Seurat en su guía (*Stuart and Butler et al*, 2019) y están amoldados para que el volumen de datos se amolde a las posibilidades técnicas de ejecución en mi ordenador personal.

Normalización de los datos

Una vez retiradas las células que no deseamos incluir en el análisis, el siguiente paso es normalizar los datos. La normalización que se aplica en este caso es una normalización mediante transformación logarítmica, ya que en casos como este, en el que todos los valores de la expresión son positivos, aplicar esta transformación ayuda a normalizar los datos. De esta forma, tendremos los valores de expresión normalizados para cada célula respecto a la expresión total.

Para realizar este paso, primero vamos a separar nuestro objeto mediante el identificativo que asignamos al crear la estructura de datos Seurat Object. De este modo, nuestros datos quedarán separados por su origen (covid 1, covid 2, control 1, control 2, …). Utilizamos la función NormalizeData de Seurat para llevar a cabo la normalización de los datos. Esta función ejecuta por defecto la normalización mediante transformación logarítmica con un factor de escalado de 10.000.

Identificación de genes altamente variables

El siguiente paso una vez se han normalizado los datos es identificar aquellos genes que presentan mayor variabilidad *cell-to-cell*, es decir, aquellos que se expresan altamente en algunas células y muy poco en otras. Se ha encontrado (*Brennecke et al*, 2013) que centrarse en estos genes ayuda a identificar y remarcar diferencias biológicas en este tipo de datos.

Para este paso, se utiliza la función de Seurat *FindVariableFeatures*, la cual calcula qué genes presentan una mayor diferencia en expresión aplicando un análisis de *mean-variance* o diferencia media. Este análisis se utiliza para tomar decisiones en investigación y se basa en tomar un riesgo, en este caso seleccionar únicamente una porción de genes a tener en cuenta, a cambio de obtener un resultado mejor o más preciso.

Esta función retorna un conjunto de 2000 genes, aquellos con mayor variabilidad de expresión, que serán utilizados en pasos posteriores de este análisis. Al estar aplicando esta función sobre el objeto que previamente habíamos separado mediante los

identificadores, esta función nos retornará los 2000 genes con mayor variabilidad para cada una de las muestras (covid 1, control 1, …).

Con el objetivo de esclarecer lo que tratamos de conseguir con este paso, el *Gráfico 2* muestra un gráfico correspondiente a una de las muestras, en este caso el paciente número 4 con COVID-19. En él podemos observar cómo, de 12175 recuentos o señales que presenta la muestra, únicamente utilizaremos en los siguientes análisis los 2000 que presentan una mayor varianza. En el gráfico mostrado en el lado derecho se puede observar el mismo gráfico con los nombres de los 15 genes que presentan mayor variabilidad. Este es meramente ilustrativo ya que en pasos posteriores del análisis entraremos más a fondo en estos genes.

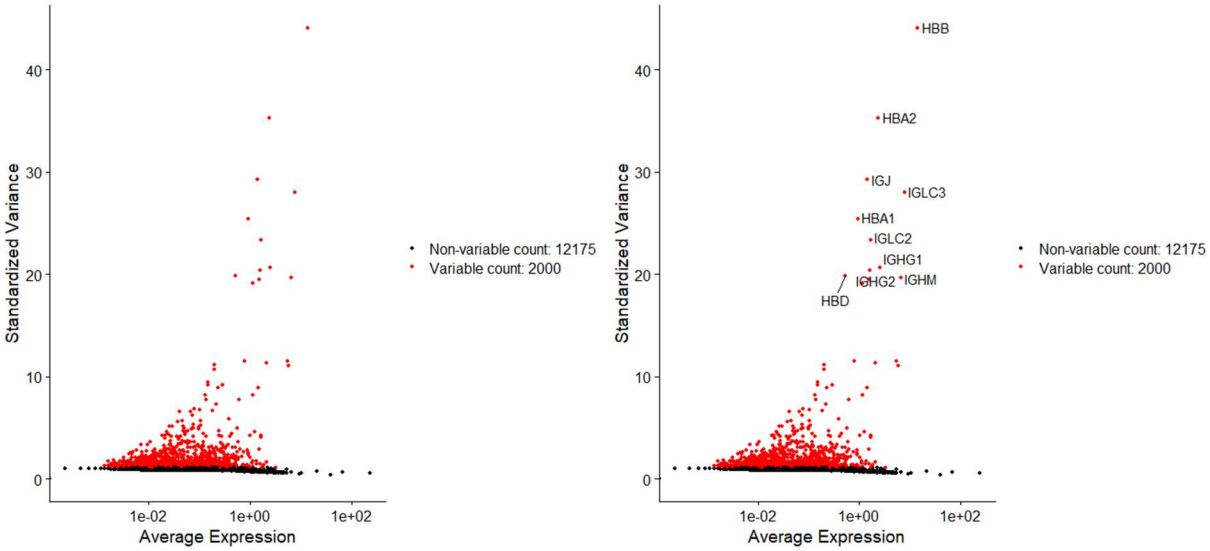

Gráfico 2. Representación de la varianza estándar de los genes de la muestra identificada como 'covid 4'.

Reintegración de los datos

Ahora que tenemos los datos normalizados y hemos escogido los 2000 genes con mayor variabilidad de cada muestra, el siguiente paso es volver a integrar todos los datos que habíamos separado por muestra en uno solo.

Para ello, utilizamos dos funciones: la primera, *FindIntegrationAnchors*, es una función que busca los puntos de similitud o de integración entre dos o más objetos. Para ello, utiliza los 2000 genes que hemos especificado en el apartado anterior, y retorna un objeto que será utilizado por la siguiente función, *IntegrateData*, la cual se encargará de juntar los 13 objetos correspondientes a las 13 muestras de nuestro estudio en uno solo mediante los puntos de integración que proporciona *FindIntegrationAnchors*. Cabe destacar que no podemos utilizar la misma función que en el inicio del análisis, la función *merge*, precisamente porque queremos que la integración de los objetos esta vez se

realice teniendo en cuenta los 2000 genes con mayor variabilidad de expresión, no todo el conjunto de genes como hace la función *merge*.

Agrupación de los datos según la condición

El resultado del paso anterior nos permite volver a tener todos los datos integrados y relacionados mediante los genes que nos interesan. Estos datos están identificados, como ya se ha dicho, por la muestra de la que provienen; sin embargo, ahora es el momento de crear una nueva columna en nuestra estructura de datos que nos permita identificar cada gen por la condición del paciente al que pertenece, es decir, pasar de covid 1, covid 2, control 1, …, a simplemente 'covid' y 'control'. Es por ello por lo que creamos una nueva columna en la que todos aquellos genes que pertenezcan a pacientes con COVID-19 serán etiquetados como 'covid' y aquellos que están sanos tendrán la etiqueta de 'control'. Este paso es necesario para que, al final del análisis, podamos especificar que queremos comparar los genes de los pacientes infectados vs los pacientes sanos.

Escalar los datos

El siguiente paso en nuestro análisis es escalar los datos. La función *ScaleData* de Seurat se utiliza en este momento y lo que hace es desplazar la expresión de los genes, de modo que la expresión media entre células sea 0, y escala la expresión de cada gen para que la varianza entre células sea de 1. De este modo, se consigue una equidad en la expresión de los genes para que aquellos genes que estén altamente expresados no dominen y sobresalgan en exceso del resto, cosa que haría que en posteriores análisis como PCA o UMAP, los resultados fueran muy extremos y pudiéramos hacernos una idea equivocada de lo que realmente ocurre.

Reducción dimensional linear

Con los datos ya correctamente escalados, vamos a llevar a cabo una reducción dimensional linear. Esto consiste en reducir el número de variables aleatorias que hay en un conjunto de datos, es decir, pasar de un conjunto más extenso a uno más reducido, en el que este espacio más reducido contiene las características que son deseadas en nuestro análisis. Básicamente este paso se realiza porque tratar con un espacio más extenso implica un mayor coste computacional, ya que los datos suelen estar muy dispersos y tratar todo el conjunto puede conllevar un tiempo de ejecución excesivo.

Es por ello por lo que en este momento llevamos a cabo un PCA (*Principal Component Analysis*). Esta técnica lleva a cabo una transformación lineal mediante la cual, un conjunto de valores, en este caso nuestros datos, son reagrupados en el espacio

dimensional en función de su varianza. Aquellos que presenten mayor varianza son colocados en el primer grupo o PC (*Principal Component*), y así sucesivamente conforme desciende la varianza. De este modo, aquellos datos más representativos e interesantes para nuestro análisis serán colocados en los primeros PCs. Esto nos permite reducir las dimensiones del conjunto de datos a analizar, ya que prescindir de aquellos que presenten menor varianza no implicará una pérdida significativa en la calidad de nuestro análisis.

Determinar la dimensionalidad del conjunto de datos

En el paso anterior llevamos a cabo un PCA con el objetivo de reducir el volumen de datos con el que trabajar, quedándonos especialmente con aquellos datos que presentan una mayor varianza y por lo tanto son más representativos del objetivo del estudio. El paso que le sigue es determinar concretamente donde hacemos el corte, es decir, dónde fijamos el límite entre con lo que nos quedamos y de lo que prescindimos. Para ello, Seurat nos ofrece varias posibilidades: la primera está basada en una implementación del método JackStraw, el cual se basa en determinar con precisión la relación entre un conjunto de variables genómicas y la agrupación en PCs (*Chung, N. C., & Storey, J. D.*, 2015). Sin embargo, debido a la poca experiencia en este método y que, personalmente, durante las prácticas no utilicé este método, se ha decidido seguir la segunda posibilidad, la cuál es igualmente válida.

La segunda posibilidad es establecer un ranking de los componentes principales (PC) basado en el porcentaje de varianza expuesto por cada uno. Esto se hace mediante la función *ElbowPlot*, la cual genera un gráfico en el que la variable independiente es la varianza, mientras que en el eje X presenta los PCs. Como se puede observar en el *Gráfico 3,* ya a partir del PC número 10, la varianza comienza a estabilizarse en un valor bajo; sin embargo, se ha decidido establecer el límite en 15, con el objetivo de perder la mínima cantidad de información posible. Por lo tanto, podemos concluir que los PCs más significativos y con los cuales nos quedaremos para continuar el análisis son del 1 al 15.

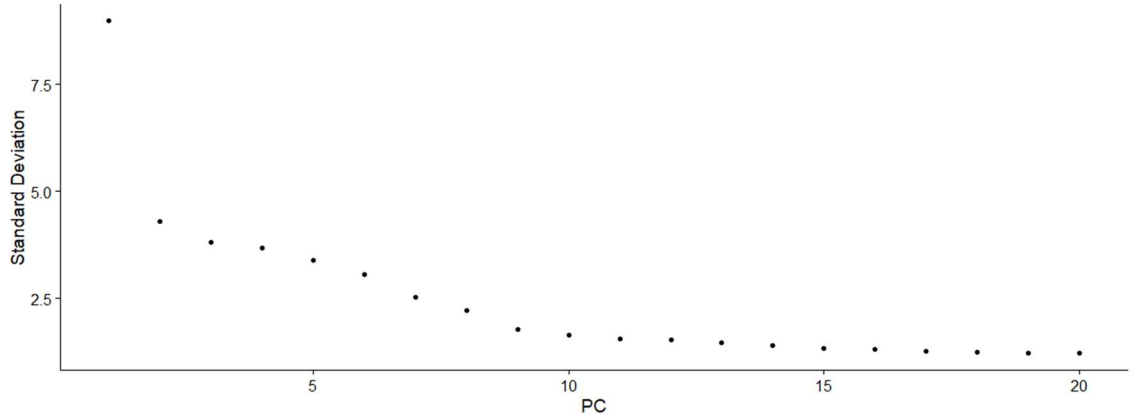

Gráfico 3. ElBow Plot, representando la desviación estándar de los PCs.

Reducción dimensional no lineal

Tras haber establecido la dimensionalidad de nuestro conjunto de datos, tenemos a nuestra disposición técnicas de reducción dimensional en este caso no lineales, que nos permiten visualizar y explorar los datos de una manera más clara. Las posibilidades que ofrece Seurat son t-SNE (*t-distributed Stochastic Neighbor Embedding*), la cual es una integración estocástica t-distribuida de proximidad, un método estadístico para la visualización de datos en grandes dimensiones; y UMAP (*Uniform Manifold Approximation and Projection*), la cual es una técnica similar a la anterior pero que también es utilizada para reducciones dimensionales no lineales en general. Una vez más, se ha optado por usar UMAP ya que es aquella con la que he trabajado en las prácticas y conozco mejor. Esta herramienta nos permitirá visualizar los clusters que generaremos a continuación de una manera más clara para el ojo humano.

Para ejecutarla en nuestro programa, utilizaremos la función de Seurat *RunUMAP*, especificando que nuestros datos se encuentran actualmente reducidos por PCA y estableciendo el número de dimensiones, es decir, de PCs a tener en cuenta, a 15 tal y como determinamos en el apartado anterior.

Creación de los clústeres

Una vez está preparado el conjunto de datos para ser visualizado mediante la reducción UMAP, llega el momento de agrupar los genes, las variables, para dar lugar a los clústeres. Un clúster es una agrupación de puntos en un espacio dimensional en el cual cada uno de esos puntos guarda una relación con el resto de puntos del mismo clúster.

Seurat nos ofrece una herramienta para construir un grafo KNN (*K-Nearest Neighbors*) basado en la distancia euclídea en el espacio del PCA. Primero de todo, los grafos KNN son herramientas muy usadas en *data mining* o minería de datos, y se basa en la premisa de que dos vértices del grafo estarán conectados por una arista si la distancia

entre esos dos puntos está entre las ‘k’ distancias más pequeñas (Yue Leng). En este caso, se utiliza la distancia euclídea, aunque también es válida cualquier distancia matemática. Esta tarea la ejecuta la función de Seurat *FindNeighbors*, a la cual también le especificaremos que utilice las 15 dimensiones que establecimos con anterioridad. De este modo, obtendremos todo un grafo de distancias entre nuestros datos, el cual utilizará la siguiente función de nuestro análisis para agrupar esos datos en clústeres.

La función que sigue a este anterior paso es *FindClusters*. Esta función de Seurat utiliza el grafo de distancias calculado por *FindNeighbors* para realizar un proceso iterativo de agrupación de los datos, colocando juntos aquellos que presenten menor distancia. Esta función cuenta con un parámetro, ‘resolution’, el cual nos permite ajustar la granularidad de los clústeres que se van a originar, es decir, nos permite especificar la distancia a la que dos puntos van a ser considerados como ‘similares’ y por lo tanto agrupados en el mismo clúster, o ‘diferentes’ y por lo tanto los colocaremos en distintos clústeres. Por supuesto, esto implica que, con valores menores de resolución, obtendremos menos clústeres, mientras que poner un parámetro muy alto de resolución nos originará muchos más clústeres. Es tarea del investigador que realiza el análisis establecer un valor para la resolución, el cuál dependerá exclusivamente de los datos que estemos tratando. De esta forma, se debe determinar un valor que sea un fiel reflejo de la realidad: ni muy pequeño porque eso daría lugar a clústeres que están compuestos por varios tipos celulares, ni muy grande porque podría hacer imposible la identificación celular.

Para visualizar los clústeres que se han construido, disponemos de la función *DimPlot*, la cual genera un gráfico como el *Gráfico 4*. Experimentalmente, a base de realizar distintas pruebas, se ha establecido el valor de la resolución a 0,5 ya que, como se observa en el *Gráfico 4*, los clústeres se encuentran bien definidos, haciendo un total de 17 clústeres.

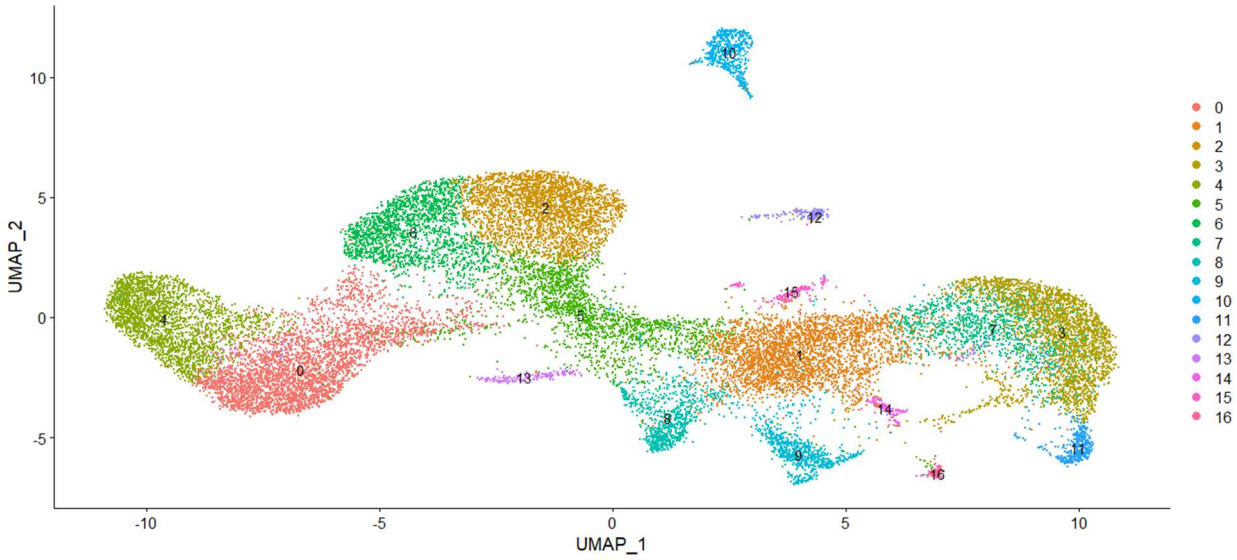

Gráfico 4. Visualización de los clústeres mediante reducción UMAP.

Identificar marcadores celulares conservados

En este momento tenemos 17 clústeres definidos en nuestro conjunto de datos. Cada uno de ellos está compuesto por una serie de genes que guardan una relación entre ellos. La tarea que tenemos ahora por delante es identificar las características de los clústeres y constatar si hemos decidido correctamente el valor de la resolución a la hora de construirlos. Para ello, vamos a utilizar las herramientas que nos proporciona Seurat con el objetivo de definir los clústeres mediante la expresión diferencial, es decir, vamos a encontrar aquellos genes que se expresan más en un determinado clúster respecto al resto, vamos a encontrar los biomarcadores de cada clúster.

La premisa sobre la que se fundamenta este proceso es la siguiente: si en un clúster concreto encontramos una fuerte expresión de genes que únicamente se expresan en un determinado tipo celular, podemos concluir que los genes incluidos en ese determinado clúster corresponden a ese tipo celular, por lo que podríamos decir que el clúster número 'x' representa a las células 'y'. Seurat nos proporciona la función *FindConservedMarkers*, la cual compara la expresión de los genes en un determinado clúster que se le especifica por parámetro a la función con respecto el resto de clústeres. De esta forma, aquellos que mayor diferencia arrojen serán los genes que predominan en el clúster en concreto. Además de ello, podemos especificar a la función a través de la variable '*grouping.var*' que queremos que realice esta comparación independientemente de la condición de cada muestra, es decir, que comparará tanto los genes de las muestras provenientes de pacientes con COVID-19 como de los pacientes sanos. Esto lo especificamos porque en este momento lo que buscamos es identificar a qué tipo celular se corresponde cada clúster; si se diera el caso que pacientes con la

enfermedad no expresaran determinados genes o que ciertos tipos celulares vieran su actividad disminuida por la enfermedad, tener en cuenta esta diferencia de condiciones podría originar resultados que no se correspondiesen con la realidad y podría dificultar la identificación celular.

Con estos datos para cada clúster, el siguiente paso es realizar una búsqueda de estos genes que hemos obtenido para asociarlos con el tipo celular en el que predominan y así poder etiquetar cada clúster con el tipo celular que representa. Esta tarea se ha realizado buscando en una base de datos de genes y proteínas como es ProteinAtlas (https://www.proteinatlas.org/). En esta página web, basta con buscar el nombre del gen para que aparezca la información relativa a él, y en concreto podemos observar en qué tipos celulares se ha encontrado expresión. Cabe destacar que, pese que la función *FindConservedMarkers* arroja como resultado una gran cantidad de genes para cada clúster, no es necesario realizar la búsqueda en bases de datos de todos ellos. Se prioriza aquellos genes que tienen una mayor expresión con respecto al resto de clústeres, a la vez que se miran los genes que resultan más identificativos por propia experiencia del investigador que lo realiza, en este caso yo basándome en mi experiencia durante las prácticas.

Para realizar la búsqueda en proteinatlas.org, se introduce en la barra de búsqueda el gen y una vez en la ficha del gen, como los datos con los que trabajamos pertenecen a células PBMC, se mira la expresión en la sangre, concretamente en la sección “RNA blood cell type specificity”. En la página se muestra un gráfico de barras con la expresión detectada en distintos tipos celulares. La mayoría de genes se expresan en diversos tipos celulares, debido a que la mayoría provienen del mismo linaje celular. Sin embargo, buscando aquellos genes más específicos de cada tipo celular y poniendo en conjunto los resultados arrojados por la búsqueda de varios genes por clúster, podemos concluir con suficiente exactitud el tipo celular que se corresponde con cada clúster.

De esta forma, podremos estudiar los efectos de la enfermedad en cada tipo celular en concreto, lo que es el objetivo principal de este proyecto.

Por coherencia de la estructura del trabajo, los resultados metodológicos de este proceso se detallan en el apartado de Resultados y discusión. Se ha seguido la metodología descrita anteriormente para identificar cada uno de los tipos más representativos de cada clúster, utilizando los genes que se muestran en la *Tabla 1* de la sección de resultados y discusión.

Una vez realizada la identificación, renombramos los clústeres, que teníamos identificados con números, con los tipos celulares. Al nombrar dos o más clústeres con el mismo identificador, como en el caso de los clústeres 1 y 7 o 12 y 15 (comentado en detalle en la sección de Resultados y discusión), pasarán a ser considerados como un único clúster. Podemos observar el resultado en el *Gráfico 5*, en el apartado de resultados y discusión.

Expresión diferencial en pacientes COVID-19 vs pacientes sanos

El último cambio que vamos a hacer a nuestros datos antes de conseguir los resultados finales del análisis es crear una columna para cada gen que tenga la información tanto del tipo celular al que pertenece como el origen de la muestra. Un ejemplo de entrada de esta columna sería: “Natural Killer – healthy”. De esta forma, podremos especificar a la función *FindMarkers* de Seurat que queremos comparar la expresión de los genes del mismo tipo celular pero entre distintas condiciones. Ejecutamos esta función comparando cada uno de los tipos celulares identificados y guardamos los resultados en un Excel. Los resultados de la función y, por tanto, lo que aparece en el Excel, tiene el siguiente formato:

Gen	p_valor	logFC media	pct 1	pct 2	p_valor ajustado

Donde:

- Gen: el nombre del gen (p.e: LYZ)
- p_valor: p_valor no ajustado
- logFC media: Fold-change, mide cuánto cambia la expresión media entre los dos grupos, en este caso entre Covid y control. Los valores positivos indican que el gen está más expresado en el grupo Covid.
- pct 1: El porcentaje de las células donde el gen es detectado en el grupo Covid.
- pct 2: El porcentaje de las células donde el gen es detectado en el grupo control.
- p_valor ajustado: p-valor ajustado según la corrección de Bonferroni (McDonald, 2014) utilizando todos los genes del conjunto de datos.

Con estos resultados finaliza el análisis de expresión diferencial, de modo que hemos obtenido datos de la expresión de una gran variedad de genes en células PBMC en pacientes con Covid-19 y en pacientes sanos que ejercen como control.

Resultados y discusión

Biomarcadores conservados y asignación de clústeres

Primero de todo expongo los resultados correspondientes a la identificación de marcadores celulares conservados para llevar a cabo la asignación de cada clúster al tipo celular más representativo de este. En la *Tabla 1* se pueden observar los genes utilizados para la identificación y el tipo celular al que se ha asignado cada clúster. A continuación de la *Tabla 1*, se detalla algunas decisiones que se han tomado para hacer la asignación de una manera lo más consecuente con los resultados obtenidos.

Tabla 1. Relación entre los clústeres, los genes utilizados para su identificación y el tipo celular asignado a cada clúster.

Clúster	*Genes utilizados para la identificación*	*Tipo celular*
0	CCL5, GZMH, CD8A, GZMA, CD8B	Linfocito T CD8
1	S100A8, S100A9, LYZ, VCAN	Monocito Clásico
2	CCR7, TCF7, IL7R, MAL, TRABD2A, LEF1	Linfocito T CD4 Primitivo
3	CPVL, CST3, CD14, FCN1, KLF4	Monocito Intermedio
4	PRF1, GNLY, IL2RB, NKG7, KLRF1	Natural Killer
5	PLEK, LCP1, ITGB2, ARPC2	No concluyente*
6	CD28, CD4, RCAN3, TRIB2, AQP3	Linfocito T CD4
7	S100A8, S100A9, LYZ, VCAN, CD14	Monocito Clásico
8	TNFRSF17, POU2AF1, TNFRSF13B, CPNE5	Linfocito B de Memoria
9	MS4A1, FCRL1, CD79A, CD22, CD19	Linfocito B Primitivo
10	G0S2, CXCR2, MMP25, TNFRSF10C, CSF3R	Neutrófilo
11	CDKN1C, MS4A7, CSF1R, TCF7L2, FMNL2	Monocito No Clásico
12	SDPR, GNG11, TUBB1, TAL1, ITGB3	Basófilo
13	TPX2, MKI67, TYMS, TOP2A, DTL	Linfocito T Regulador
14	HBA1, HBD, HBA2, SNCA, HBB	Eritroblasto
15	CLC, GATA2, FAM101B, AKAP12	Basófilo
16	ITM2C, SERPINF1, PTPRS, LILRA4, PLD4	Célula Dendrítica

*El clúster número 5 se encuentra en una zona central del *Gráfico 4*, cuyo centro podríamos ubicar entorno al punto (0,0). El resultado de la búsqueda de marcadores

celulares en este clúster ha evidenciado que está compuesto por genes con especificidad baja, genes que se encuentran en todo tipo de células del PBMC y por tanto no son marcadores de ningún tipo o linaje celular concreto. Este resultado es fruto de la aproximación que se lleva a cabo con el valor de la resolución a la hora de definir los clústeres. Es por ello por lo que se ha decidido descartar este clúster del análisis final debido a que no daría resultados que se correspondiesen a ningún tipo celular determinado.

El clúster número 7 ha sido catalogado como monocito clásico ya que, pese a contener genes identificativos tanto de monocitos clásicos como intermedios, tiene predominancia por los primeros, por lo que pasará a contarse los clústeres 1 y 7 como uno solo, correspondiente a monocito clásico. Esto ocurre porque muchas veces las fronteras entre un tipo y otro no están tan bien definidas; sin embargo, cabe tener en cuenta que las diferentes clases de monocitos provienen del mismo linaje celular, por lo que pese a ser consideradas diferentes y analizadas por separado, se hará una visión general concerniente a los monocitos.

Los clústeres 12 y 15 han sido categorizados como basófilos. Pese a encontrarse separados, el análisis de sus genes más conservados ha evidenciado la predominancia de los genes pertenecientes a basófilos. La separación que se observa en el *Gráfico 4* se debe a que el clúster 12 contiene algunos genes que también se encuentran en neutrófilos, por ello está más cercano al clúster 10, mientras que el clúster 15 presenta algunos genes que también aparecen en eosinófilos. Pese a todo, al igual que se ha comentado en el anterior párrafo, los basófilos, neutrófilos y eosinófilos provienen del mismo linaje celular de granulocitos, por lo que también se hará una visión general englobándolos como granulocitos.

Con el renombramiento y agrupación de los clústeres, podemos visualizar el gráfico UMAP con los tipos celulares identificados. Esto nos da una visión mas intuitiva de las diferencias y similitudes entre tipos celulares. Se puede observar, por ejemplo, que los linfocitos y NK se encuentran en la parte izquierda del *Gráfico 5* y cercanos entre sí, mientras que los monocitos están en el otro extremo del gráfico o los neutrófilos y basófilos más separados del resto.

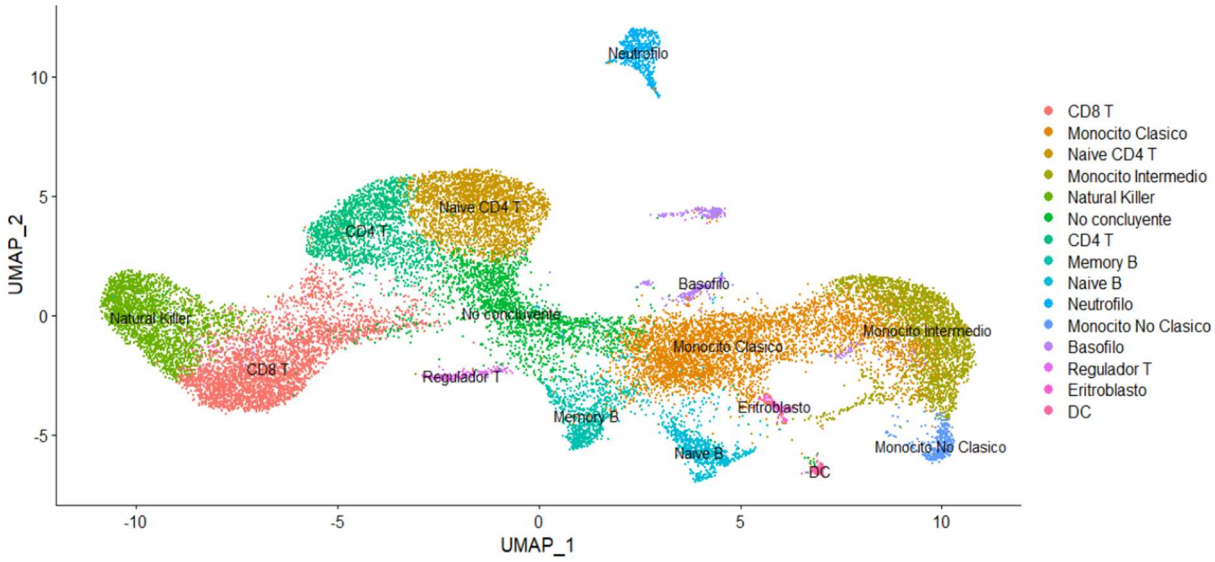

Gráfico 5. Visualización de los clústeres identificados con el tipo celular que representan.

Expresión diferencial entre pacientes COVID-19 y pacientes sanos

Del grueso de genes que las funciones utilizadas dan como resultado, se debe hacer un proceso de investigación y selección de aquellos que resultan más relevantes o que tienen una implicación más directa con lo que se está evaluando. En la Tabla 2, se encuentran los genes con mayor diferencia de expresión, indicando en qué tipo/s celular/es se encuentran y una pequeña descripción de su función, y que a su vez tienen relevancia en el análisis. Para poner un ejemplo de qué genes no tiene sentido comentar, en algunos tipos celulares se ha encontrado la presencia de genes como XIST o PLAC8, los cuales son genes que regulan aspectos relacionados con el género y que se expresan únicamente en mujeres y no en hombres (XIST es un gen que participa en la inactivación de un cromosoma X en las mujeres, y PLAC8 es un gen asociado a la placenta). Estos aparecen como genes con diferencia de expresión, pero es debido a que en un grupo tal vez haya más mujeres que en el otro o que incluso en uno de ellos no haya mujeres. Este es un ejemplo de genes que no se deben tener en cuenta en la discusión de resultados debido a que no aportan información sobre la condición estudiada, que es la enfermedad del COVID-19.

El proceso de búsqueda de las funciones de los genes se ha realizado en la base de datos *GeneCards, The Human Gene Database* (https://www.genecards.org/), la cual muestra a su vez los resultados de otras bases de datos como *UniProt* (https://www.uniprot.org/).

La *Tabla 2* muestra los distintos genes junto con la condición en la que se encuentran sobreexpresados (COVID-19 o control), una pequeña descripción de su función y el tipo o tipos celulares en los que aparecen. Además, se encuentran agrupados por función o tipo para organizar la tabla y exponer los grupos que se van a discutir a continuación.

Tabla 2. Localización, sobreexpresión y descripción de los genes con diferencia de expresión más relevantes.

Gen/es	Sobreexpresión en	Descripción	Tipo celular
Proteínas de señalización y activación			
TYROBP	Control	Inmunoreceptor que inhibe las funciones de lisis celular por parte de linfocitos T CD8 y NK	Linfocitos T CD8, NK
IL32	COVID-19	Miembro de la familia de las citoquinas, las cuales participan en la respuesta inmune innata y adaptativa. Induce la expresión de TNF por parte de macrófagos	Linfocitos T CD8, NK, Basófilos
IL7R	COVID-19	Receptor de la IL7. La IL7 es una citoquina necesaria para la supervivencia y expansión de los linfocitos.	Linfocitos T CD8
IL3RA	Control	Receptor de la interleucina 3, la cual es un factor estimulante de colonias de granulocitos y macrófagos	Basófilos
SOCS3	COVID-19	Es un supresor de la señalización de citoquinas, tiene funciones reguladoras en la respuesta mediada por linfocitos T CD4	Linfocitos T CD4, Linfocitos T CD4 primitivos
VCAN	COVID-19	Participa en la señalización intercelular y en la conexión de células con la matriz extracelular	Monocitos Clásicos
ISG15	COVID-19	Proteína inducida por interferón que presenta múltiples funciones como actividad quimiotáxica hacia neutrófilos, señales intercelulares o actividad antiviral	Células dendríticas
Inmunoglobulinas			
IGLC3, IGHG1, IGHM, …	COVID-19	Regiones de las cadenas de las inmunoglobulinas	Linfocitos T CD8, Linfocitos T CD4, Linfocitos T CD4 primitivos, NK, Linfocitos B de memoria, Linfocitos B primitivos, Monocitos Clásicos
Proteínas anti-antiapoptóticas			
XAF1	COVID-19	La proteína codificada por este gen es un regulador negativo de las proteínas inhibidoras de la apoptosis	Linfocitos T CD8, NK, Células dendríticas, Neutrófilos
BOK	Control	Proteína con funciones anti-apoptóticas	Linfocitos T Reguladores
MTRNR2L1	COVID-19	Factor antiapoptótico	Linfocitos B primitivos
Factores de transcripción			
MYC	COVID-19	Factor de transcripción que activa la transcripción de genes involucrados en el crecimiento	Linfocitos T CD4
STAT1	COVID-19	Transductor de señales y activador de la transcripción que media las respuestas celulares a los interferones y citoquinas.	NK
IKZF1	COVID-19	Factor de transcripción. Participa en el desarrollo de linfocitos.	Linfocitos T Reguladores

XBP1	COVID-19	Factor de transcripción que regula los genes del MHC tipo II	Linfocitos B de memoria
Proteínas inducidas por interferón (IFN)			
IFI27	COVID-19	Proteína inducida por el interferón alpha y que presenta actividad antiviral	Linfocitos B de memoria, Monocitos Clásicos, Monocitos Intermedios, Monocitos No clásicos, Células Dendríticas, Eritroblastos
IFI6	COVID-19	Proteína inducida por el interferón alpha y que regula negativamente el flujo de señales de la apoptosis	Monocitos Clásicos
IFIT3, IFIT2	COVID-19	Proteínas inducidas por interferón que actúan como inhibidoras de procesos celulares y virales, sobre todo inhiben la expresión y replicación de mRNAs virales	Neutrófilos
IFITM3	COVID-19	Proteína inducida por interferón y que inhibe la entrada de virus a al citoplasma de la célula huésped. Se coloca en la membrana.	Monocitos Intermedios, Monocitos No clásicos, Neutrófilos
IFI44L	COVID-19	Presenta actividad antiviral	Linfocitos T CD4, NK, Células Dendríticas
MX1	COVID-19	La proteína que codifica es inducida por interferón e inhibe la replicación de diferentes virus de DNA y RNA	Linfocitos T CD8, Linfocitos T CD4, NK, Monocitos Clásicos
RSAD2	COVID-19	Proteína inducible por interferón que participa en la respuesta celular antiviral y la señalización inmunitaria innata	Neutrófilos
Ciclo celular, proliferación y diferenciación			
NFKBIZ	COVID-19	Está involucrado en la diferenciación de linfocitos T CD4	Linfocitos T CD4 primitivos
TSC22D3	Control	Inhibe la diferenciación celular	Linfocitos T CD4 primitivos
CD69	COVID-19	La expresión de la proteína que codifica es inducida tras la activación de los linfocitos T y participa en la proliferación celular y reconocimiento.	Linfocitos T CD4
PIM1	COVID-19	Quinasa que favorece la proliferación y supervivencia celular	Linfocitos T CD4, Linfocitos T CD4 primitivos
TXNIP	Control	Es un represor transcripcional. La sobreexpresión de este puede provocar arrestos en la fase G0/G1 del ciclo celular	Linfocitos T CD4 primitivos
CDKN1C	Control	Es un regulador negativo de la proliferación celular e inhibe complejos que hace pasar la célula de la fase G1 a la S del ciclo celular	Monocitos No clásicos
ARPC1B	Control	Componente de un complejo que media la polimerización de actina, aunque también presenta funciones de regulación de la transcripción	Células dendríticas

		y de reparación de DNA dañado	
Proteínas de membrana y transporte			
CLIC3	Control	Forma canales transmembranales para el ion cloro	NK
CYB561D2	Control	Citocromo que cataliza una reducción de agentes quelantes	Linfocitos T Reguladores
STOML1	Control	Proteína de membrana que realiza funciones de transporte transmembrana	Linfocitos T Reguladores
IFT43	Control	Componente del complejo de transporte intraflagelar	Linfocitos T Reguladores
BANK1	COVID-19	Proteína involucrada en la movilización del calcio intracelular	Linfocitos B primitivos
Chaperonas y procesamiento de proteínas			
HSP90B1, HSPA5	COVID-19	Chaperonas que participan en el transporte y procesamiento de proteínas que son secretadas	Linfocitos B de memoria
CALR	COVID-19	Chaperona que ayuda en el plegamiento de proteínas en el RE pero que también se encuentra en el núcleo, lo que sugiere que tiene un rol en la transcripción también.	Linfocitos B de memoria
CLU	COVID-19	Chaperona que previene la agregación de proteínas no nativas y protege a la célula contra la apoptosis y citólisis.	Monocitos Intermedios
NAP1L1	Control	Chaperona de la histona que participa en la replicación y reparación del DNA	Monocitos No clásicos
SEL1L3	COVID-19	La proteína codificada por este gen es parte de un complejo que participa en la degradación de proteínas mal plegadas y en la maduración y secreción de LPL	Linfocitos B primitivos
Inflamación y respuesta inmune en general			
S100A8, S100A9, S100A12	COVID-19	Familia de las proteínas S100 que ejerce múltiples funciones en los procesos de inflamación y respuesta inmunitaria, como el reclutamiento de linfocitos, la inducción de producción de citoquinas, etc.	Monocitos Clásicos, Monocitos Intermedios, Monocitos No clásicos, Neutrófilos, Basófilos
FCER1A	Control	Es el receptor de las inmunoglobulinas E, iniciadoras de la respuesta alérgica	Basófilos
MZB1	COVID-19	Promueve el ensamblaje y secreción de IgM. Ayuda a diversificar las funciones periféricas de los linfocitos B	Linfocitos B de memoria
BIRC3	COVID-19	Proteína multifuncional que principalmente inhibe la apoptosis, pero también regula señales inflamatorias e inmunitarias.	Linfocitos B primitivos
SWAP70	COVID-19	Regula los procesos esenciales para la entrada de los linfocitos B en los ganglios linfáticos.	Linfocitos B primitivos

SELL	COVID-19	Molécula de adhesión superficial, facilita la migración de leucocitos hacia los órganos linfoides secundarios y las zonas con inflamación	Monocitos Intermedios, Neutrófilos
Variantes alélicas HLA			
HLA-D*	Control	Variante alélica de la familia de las proteínas HLA, las cuales forman el MHC. Esta variante en concreto forma parte del MHC de clase II.	Monocitos Clásicos, Monocitos Intermedios
HLA-A*	COVID-19	Variante alélica de la familia de las proteínas HLA, las cuales forman el MHC. Esta variante en concreto forma parte del MHC de clase I.	Monocitos Clásicos
HLA-C	Control	Variante alélica de la familia de las proteínas HLA, las cuales forman el MHC. Esta variante en concreto forma parte del MHC de clase I.	Neutrófilos
Funciones metabólicas			
PSAP	Control	Preproteina que se procesa en 4 productos, todos con funciones metabólicas	Células dendríticas
UBIAD1	Control	Transferasa que participa en el metabolismo de fosfolípidos	Linfocitos T Reguladores
CKB	Control	Cataliza la transferencia de fosfato del ATP a la creatina fosfato.	Monocitos No clásicos
Proteínas específicas			
CD8A	COVID-19	Codifica la cadena alfa de la glicoproteína CD8	Linfocitos T CD8
GNLY	Control	Proteina presente en los gránulos citotóxicos de los linfocitos T citotóxicos y NK que mata patógenos intracelulares	Linfocitos T CD8, NK
Eritroblastos			
ALAS2	COVID-19	Es una enzima específica de eritroblastos, la cual cataliza el primer paso de la biosíntesis del grupo hemo	Eritroblastos
TRIM58	COVID-19	Ligasa cuya expresión es inducida en las últimas fases de la eritropoiesis	Eritroblastos
AHSP	COVID-19	Chaperona que se une específicamente a la alpha-globina libre y participa en la unión de la hemoglobina	Eritroblastos
GLRX5	COVID-19	Es requerida para la correcta regulación de la síntesis de hemoglobina	Eritroblastos

Proteínas de señalización y activación

Cuando se da una infección, el intercambio de señales entre las distintas células del sistema inmune juega un papel clave en la respuesta, ya que se necesita toda una cascada de moléculas para promover la proliferación y migración de los linfocitos hacia los tejidos infectados, inducir la expresión y secreción de moléculas efectoras, etc.

Una gran parte de esta señalización es mediada por las citoquinas y las interleucinas. Las citoquinas son proteínas de bajo peso molecular responsables de la comunicación intercelular, uniéndose a receptores específicos favoreciendo la proliferación y diferenciación, así como la quimiotaxis. Las interleucinas son un conjunto de citoquinas y ambas son producidas por los leucocitos.

Se puede observar en los resultados de la *Tabla 2* cómo en líneas generales la expresión de interleucinas y receptores de estas (IL32, IL7R) así como de otras proteínas con funciones similares (VCAN, ISG15, SOCS3) se encuentra incrementada en el grupo COVID-19. Esto concuerda con la activación de la respuesta inmune en el sistema infectado y el correspondiente aumento en la expresión de proteínas involucradas en la transducción de señales.

Casos que merece la pena comentar en este 'subgrupo' son los genes TYROBP o la subunidad alpha del receptor de IL3, ya que presentan mayor expresión en el grupo control. El TYROBP o DAP12 es un inmunoreceptor con actividad en el dominio citosólico de la membrana celular. Este se asocia con el KIR, otro inmunoreceptor e inhibe la lisis celular por parte de linfocitos citotóxicos y NK. Por ello, le encuentro sentido a que este se sobreexprese en el grupo control, ya que regula la acción de estas células, mientras que en la respuesta inmune esté infraexpresado para así que la acción de las células efectoras se dé en mayor medida.

Por otro lado, la infraexpresión de IL3RA en el grupo COVID-19 no me cuadra con la respuesta inmune que se da en la infección. Sin embargo, la unión de la IL3 al receptor de esta depende de la subunidad B del receptor, por lo que puede que no afecte en esto.

Inmunoglobulinas

Todos los genes estructurales de las inmunoglobulinas se encuentran sobreexpresados en los pacientes con COVID-19 (IGLC3, IGHG1, IGHM, IGHG4, …), lo cual denota que en estos pacientes se ha producido una activación de los linfocitos B, que son los que las sintetizan, así como de las distintas células del sistema inmune debido a la infección, mientras que en los pacientes sanos no se da esta activación.

Proteínas anti-antiapoptóticas

El SARS-CoV-2 tiene como sello identificativo la promoción de la apoptosis celular (*Ren et al*, 2020) y se ha demostrado que mata los linfocitos y por lo tanto inmunosuprime al huésped (*Laterre et al*, 2020). Es interesante observar la sobreexpresión del gen XAF1 en el grupo COVID-19. Este gen codifica una proteína que se une a un miembro de la familia de las proteínas IAP (inhibitor of apoptosis) e inhibe su acción. Como han comentado otros autores (*Zhu et al*, 2020), con este gen podemos ver los mecanismos que utiliza el virus para promover e inducir la apoptosis de las células plasmáticas del individuo, que es mediante la sobreexpresión de genes que inhiben los genes con funciones antiapoptóticas.

Proteínas inducidas por interferón (IFN)

Los interferones (IFNs) son glicoproteínas con función de señalización que son producidas por una célula infectada por un virus. El objetivo de esta es generar una respuesta antiviral en las células cercanas, las cuales detectarán estos interferones.

Estas proteínas se encuentran sobreexpresadas en el grupo COVID-19, en concordancia con la infección. La mayoría de estas pertenecen a la familia de las proteínas inducibles por interferón (IFI) y presentan funciones distintas en la respuesta frente al virus. Algunos ejemplos de estas son: IFI27, proteína inducible por interferón alpha que presenta actividad antiviral. Cabe destacar que no se registra expresión de esta proteína en el grupo control, por lo que podemos deducir que esta proteína solo se expresa en condiciones de infección viral. La proteína IFI6 inhibe el flujo de señales de la apoptósis; las proteínas IFIT2 e IFIT3 principalmente inhiben la expresión y replicación de mRNAs virales, así como la proteína MX1; la proteína IFITM3 se coloca en la membrana e impide el acceso del virus en el citoplasma celular.

Factores de transcripción

Los factores de transcripción son proteínas que se unen específicamente a una secuencia de DNA, regulando así la transcripción. La sobreexpresión de estos factores en el grupo COVID-19 implica la necesidad de la célula de sintetizar las proteínas por las que codifican estos genes en mayor medida.

Algunos de los genes que vemos sobreexpresados son: STAT1, un factor de transcripción y transductor de señales que es inducido en respuesta a citoquinas y factores de crecimiento y que sirve como factor de transcripción a una gran variedad de genes importantes en la viabilidad celular y la respuesta frente a diversos estímulos. Otros autores han apuntado que este es una de las vías de respuesta frente a la COVID-

19 (*Zhu et al*, 2020). IKZF1 es un factor de transcripción que participa en el desarrollo de linfocitos; XBP1 regula los genes que componen el MHC de tipo II.

Inflamación y respuesta inmune en general

En esta categoría he decidido incluir genes que se encuentran sobreexpresados sobre todo en monocitos, macrófagos y linfocitos B y que ejercen funciones en la respuesta a la inflamación y la respuesta inmune en general.

Los genes de la familia S100 (S100A8, S100A9, S100A12, …) se encuentran sobreexpresados en los individuos con COVID-19 en monocitos y macrófagos y se encargan de regular aspectos de la progresión y diferenciación en el ciclo celular. Estudios recientes (*Guo et al*, 2021) han denotado la implicación de la sobreexpresión de genes de esta familia con un desorden inmunitario que produce neutrófilos inmaduros aberrantes.

Otros genes sobreexpresados en el grupo COVID-19 son: MZB1 en linfocitos B, que participan en la respuesta de estos; SWAP70 y SELL, los cuales tienen funciones de promoción celular hacia los ganglios linfáticos, lugar donde realiza principalmente la presentación de antígenos; BIRC3, que realiza diversas funciones, tanto inhibidoras de la apoptosis como de regulación de señales de la respuesta inmune e inflamatoria. Como vemos, todas estas funciones están orientadas a combatir la infección y forman parte de los mecanismos del sistema inmune, por lo que su sobreexpresión en el grupo enfermo queda justificada.

Ciclo celular, proliferación y diferenciación

En este grupo de genes se encuentran aquellos que tienen un papel regulador en el ciclo celular y en la diferenciación. Algunos de estos genes se encuentran sobreexpresados en el grupo COVID-19 y otros en el grupo control. Donde se encuentre mayor expresión dependerá de la función de cada gen; así podemos observar cómo genes que inhiben la diferenciación (TSC22D3), sirven como regulador negativo de la proliferación (CDKN1C) o son represores transcripcionales (TXNIP) se encuentran infraexpresados en el grupo COVID-19, mientras que los promotores de la diferenciación celular (NFKBIZ, CD69) y la proliferación (PIM1) se encuentran sobreexpresados en el grupo COVID-19. Esto nos indica las necesidades de la célula delante de una infección de activar sus mecanismos de proliferación celular y transcripción y síntesis de proteínas para hacerle frente.

Proteínas de membrana y transporte

Las proteínas aquí agrupadas tienen en común que se encuentran ubicadas en la membrana celular o bien que realizan funciones relacionadas con el transporte. Lo interesante de estas es que, excepto el gen BANK1 involucrado en la movilización del ion calcio intracelular, el resto se encuentran sobreexpresadas en el grupo control. La justificación del gen BANK1 es que el ion calcio intracelular es utilizado como mensajero para activar funciones celulares diversas, desde la contracción o la secreción hasta la expresión de genes. Esto cuadra con la situación de estrés en la que se ve la célula durante una infección. Por otro lado, la sobreexpresión del resto de genes en el grupo control me sugiere dos hipótesis: o bien la expresión de estas proteínas es reducida para centrarse en los mecanismos para hacer frente al virus, o bien es el virus el que induce este cambio de expresión en las proteínas de la membrana para su propio beneficio. Sea cual fuere, sería necesaria una investigación más profunda en ello para esclarecer estos resultados. De esta forma, estos resultados podrían abrir una línea de investigación sobre las proteínas de membrana en las células del huésped tras la infección por SARS-CoV-2.

Funciones metabólicas

Estos genes siguen la misma línea de pensamiento que los anteriores, los genes que codifican proteínas de membrana y transporte. Encontramos 3 genes sobreexpresados en el grupo control, en distintos grupos celulares, que tienen funciones metabólicas o involucradas en procesos metabólicos. Las hipótesis para explicar estos resultados son las mismas que en la anterior ocasión, o bien se da porque la célula, ante el estrés, se centra en combatir la infección y quita recursos de otras vías metabólicas; o bien es el propio SARS-CoV-2 el que inhibe la expresión de estas, ya sea para utilizar los recursos en su beneficio o para matar a la célula.

Chaperonas y procesamiento de proteínas

Las chaperonas son un conjunto de proteínas presentes en todas las células cuya función es la de ayudar en el plegamiento y transporte durante la síntesis de proteínas. Los resultados muestran multitud de chaperonas sobreexpresadas en el grupo COVID-19. Muchos virus utilizan el sistema de chaperonas del huésped para infectar, replicar su material genético y propagarlo (*Paladino et al*, 2020). Además, muchas proteínas virales comparten epítopos altamente reconocibles con las chaperonas humanas, provocando así que anticuerpos antivirales puedan reaccionar con las chaperonas, generando una respuesta autoinmune (*Paladino et al*, 2020).

Variantes alélicas HLA

Una familia de genes que quería destacar es la familia de las HLA, que forman el complejo mayor de histocompatibilidad (MHC). Esta familia cuenta con 3 variantes alélicas de clase I, es decir, que forman el MHC de clase I y que son HLA-A, HLA-B y HLA-C. Para el MHC de clase II, lo componen las variantes HLA-DP, HLA-DQ y HLA-DR. Diversos estudios han apuntado que existe una relación entre las variantes alélicas de HLA presentes en un individuo y la susceptibilidad y mortalidad de la COVID-19 (*Lorente et al*, 2021) (*Tavasolian et al*, 2021). Los resultados que yo he obtenido de mi análisis muestran la presencia de la variante HLA-D en monocitos del grupo control, mientras que la variante HLA-A es la presente en monocitos del grupo COVID-19. Además, los neutrófilos del grupo control presentan la variante HLA-C. Sin embargo, y como también mencionan en los artículos citados, la información que se tiene sobre la influencia de las variantes de HLA en la COVID-19 no está estudiada a fondo todavía y se requieren más investigaciones para poder relacionar de forma fiable ambos aspectos.

Eritroblastos

En este último grupo he decidido incluir genes que se encuentran en los eritroblastos y que presentan funciones relacionadas con la eritropoyesis y la síntesis del grupo hemo. Estos genes se encuentran sobreexpresados en el grupo COVID-19, lo cual sugiere un incremento en la eritropoyesis en estos individuos. Un artículo publicado recientemente (*Huerga Encabo et al*, 2021) comenta que los pacientes con COVID-19 presentan un aumento en el número de células rojas nucleadas de la sangre, lo cual podría estar en consonancia con la sobreexpresión de los genes que encuentro en mi análisis.

Proteínas específicas

En esta sección he incluido dos genes que son específicos de los linfocitos T citotóxicos y los NK. El primero de ellos es el CD8A, molécula que junto con el CD8B forma el receptor de superficie de los linfocitos T CD8, una glicoproteína que media la interacción entre células del sistema inmune. La sobreexpresión de este gen se da en el grupo COVID-19 debido a la proliferación de linfocitos T citotóxicos debido a la infección.

Por otro lado, el gen GNLY se encuentra sobreexpresado en el grupo control, tanto en linfocitos T CD8 como en NK. Este gen codifica para la granulisina, una proteína situada en los gránulos citotóxicos y que es secretada cuando se identifica una célula infectada, con el objetivo de matarla. La infraexpresión de este gen en la condición de los pacientes de COVID-19 es algo que me llama la atención, ya que en la circunstancia de la infección y con la proliferación de linfocitos que hemos visto que se da, debería también promoverse la síntesis de esta proteína para matar las células infectadas. Una posible

hipótesis para justificar este hecho podría ser que, debido a que en este análisis se evalúa el mRNA, no las proteínas, se dé el caso que en el grupo COVID-19, la infección promueve la síntesis de factores de transcripción para la granulisina y por lo tanto se encuentra menos mRNA de esta en la célula, ya que gran parte de este ha pasado por el proceso de la traducción, incentivado por los factores de transcripción, para dar lugar a las proteínas citolíticas.

Conclusiones

El análisis de expresión diferencial realizado a partir de la scRNA-sequencing ha permitido conocer la expresión de una gran cantidad de genes implicados en funciones muy distintas. Además de la expresión, nos permite conocer en qué tipos celulares se expresa y bajo qué condiciones. Los resultados obtenidos evidencian la respuesta inmune que se da en los pacientes con COVID-19, a través de la sobreexpresión de inmunoglobulinas, citoquinas y genes de proliferación de leucocitos. Esta sobreexpresión era previsible, sin embargo, el estudio de qué genes en concreto son los que se sobreexpresan nos permite conocer las vías, tanto del virus como del individuo, que se desarrollan en el proceso de infección. Así mismo, también han salido a la luz resultados que podían no ser tan previsibles, como es la infraexpresión de proteínas de membrana y transporte o proteínas involucradas en reacciones metabólicas. Como se ha comentado en la discusión de los resultados, surgen diversas hipótesis para explicar estos fenómenos que pueden servir para iniciar nuevas investigaciones más concretas y suponen un aporte en materia de innovación hacia el ámbito científico.

A modo de recapitulación, se quiere exponer en líneas generales los aspectos en los que más diferencias se han encontrado en un grupo respecto al otro. Los genes que codifican para proteínas que forman parte de las Ig, las que tienen función de señalización intercelular, los genes que participan en la proliferación, crecimiento y desarrollo celular, así como sus factores de transcripción, las chaperonas que ayudan con el plegamiento y transporte de las proteínas sintetizadas, las proteínas específicas de ciertos tipos celulares como el completo CD8 y aquellas que participan en la respuesta inmune e inflamatoria en general, se encuentran sobreexpresadas en los pacientes COVID-19. Y cuando hablamos de proteínas que participan en el ciclo celular y diferenciación, podemos observar claramente como aquellas que lo promueven son sobreexpresadas en el grupo enfermo, mientras que aquellas con funciones inhibitoras de este ciclo y diferenciación son infraexpresadas. Con el proceso de apoptósis podemos observar algo interesante: en condiciones de infección, la célula expresa factores antiapoptóticos con el fin de aumentar la supervivencia de la célula; el SARS-CoV-2, como parte de su mecanismo de acción, promueve la expresión de un factor que inhibe la actividad antiapoptótica (XAF1), promoviendo de esta manera la apoptósis. En cambio, genes que codifican proteínas con dominios transmembrana y con funciones de transporte o metabolización se expresan con menor intensidad en pacientes enfermos que en pacientes sanos.

Toda esta información nos permite elaborar perfiles de expresión, caracterizar la enfermedad, sus mecanismos y respuestas, con la posibilidad de servir como base de futuras investigaciones y con el objetivo de elaborar estrategias terapéuticas específicas y efectivas. Este análisis supone una potente herramienta a la hora de determinar cómo actúa un determinado patógeno, ya que nos permite ahondar en la expresión de cada tipo celular concreto y visualizar resultados que de otro modo, llevando a cabo el análisis sobre un grueso de células o un tejido, podrían no salir a la luz.

Bibliografía

1. Wilk, A. J., Rustagi, A., Zhao, N. Q., Roque, J., Martínez-Colón, G. J., McKechnie, J. L., Ivison, G. T., Ranganath, T., Vergara, R., Hollis, T., Simpson, L. J., Grant, P., Subramanian, A., Rogers, A. J., & Blish, C. A. (2020). A single-cell atlas of the peripheral immune response in patients with severe COVID-19. *Nature medicine*, *26*(7), 1070–1076. https://doi.org/10.1038/s41591-020-0944-y
2. Stuart and Butler et al. (2019). Comprehensive Integration of Single-Cell Data. Differential expression testing. Seurat - Guided Clustering Tutorial. https://satijalab.org/seurat/archive/v3.2/pbmc3k_tutorial.html (visitado el 05/04/2021).
3. Brennecke, P., Anders, S., Kim, J. K., Kołodziejczyk, A. A., Zhang, X., Proserpio, V., Baying, B., Benes, V., Teichmann, S. A., Marioni, J. C., & Heisler, M. G. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature methods*, *10*(11), 1093–1095. https://doi.org/10.1038/nmeth.2645
4. Yue Leng. KNN-Graph. https://github.com/lengyyy/KNN-Graph (visitado el 28/04/21).
5. Stuart and Butler et al. (2019). Comprehensive Integration of Single-Cell Data. Differential expression testing. Tutorial: Integrating stimulated vs. control PBMC datasets to learn cell-type specific responses. https://satijalab.org/seurat/archive/v3.2/immune_alignment.html (visitado el 05/04/2021).
6. Colaboradores de Wikipedia. Dimensionality reduction. Wikipedia, La enciclopedia libre. https://en.wikipedia.org/wiki/Dimensionality_reduction (visitado el 25/04/2021).
7. Colaboradores de Wikipedia. Dimensionality reduction. Wikipedia, La enciclopedia libre. https://en.wikipedia.org/wiki/Principal_component_analysis (visitado el 25/04/2021).
8. Chung, N. C., & Storey, J. D. (2015). Statistical significance of variables driving systematic variation in high-dimensional data. *Bioinformatics (Oxford, England)*, *31*(4), 545–554. https://doi.org/10.1093/bioinformatics/btu674

9. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. (s.f). https://umap-learn.readthedocs.io/en/latest/ (visitado el 28/04/2021).
10. McDonald, J.H. (2014). Handbook of Biological Statistics (3rd ed.). Sparky House Publishing, Baltimore, Maryland. http://www.biostathandbook.com/multiplecomparisons.html (visitado el 03/05/2021).
11. Stuart and Butler et al. (2019). Comprehensive Integration of Single-Cell Data. Differential expression testing. https://satijalab.org/seurat/archive/v3.0/de_vignette.html (visitado el 03/05/2021).
12. Rubio, S., Pacheco-Orozco, R. A., Gómez, A. M., Perdomo, S., & García-Robles, R. (2020). DNA Next-Generation Sequencing (NGS): Present and Future in Clinical Practice: Present and future in clinical practice. *Universitas Médica*, *61*(2). https://doi.org/10.11144/Javeriana.umed61-2.sngs
13. Soon, W. W., Hariharan, M., & Snyder, M. P. (2013). High-throughput sequencing for biology and medicine. *Molecular systems biology*, *9*, 640. https://doi.org/10.1038/msb.2012.61
14. Ari Ş., Arikan M. (2016) Next-Generation Sequencing: Advantages, Disadvantages, and Future. In: Hakeem K., Tombuloğlu H., Tombuloğlu G. (eds) Plant Omics: Trends and Applications. Springer, Cham. https://doi-org.sabidi.urv.cat/10.1007/978-3-319-31703-8_5
15. Krishanpal Anamika, Srikant Verma, Abhay Jere and Aarti Desai (January 14th 2016). Transcriptomic Profiling Using Next Generation Sequencing - Advances, Advantages, and Challenges, Next Generation Sequencing - Advances, Applications and Challenges. Jerzy K Kulski, IntechOpen, DOI: 10.5772/61789.
16. Kukurba, K. R., & Montgomery, S. B. (2015). RNA Sequencing and Analysis. *Cold Spring Harbor protocols*, *2015*(11), 951–969. https://doi.org/10.1101/pdb.top084970
17. Shapiro, E., Biezuner, T., & Linnarsson, S. (2013). Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature reviews. Genetics*, *14*(9), 618–630. https://doi.org/10.1038/nrg3542

18. Tanay, A., & Regev, A. (2017). Scaling single-cell genomics from phenomenology to mechanism. *Nature*, *541*(7637), 331–338. https://doi.org/10.1038/nature21350
19. Raj, B., Wagner, D. E., McKenna, A., Pandey, S., Klein, A. M., Shendure, J., Gagnon, J. A., & Schier, A. F. (2018). Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nature biotechnology*, *36*(5), 442–450. https://doi.org/10.1038/nbt.4103
20. Levitin, H. M., Yuan, J., & Sims, P. A. (2018). Single-Cell Transcriptomic Analysis of Tumor Heterogeneity. *Trends in cancer*, *4*(4), 264–268. https://doi.org/10.1016/j.trecan.2018.02.003
21. Neu, K. E., Tang, Q., Wilson, P. C., & Khan, A. A. (2017). Single-Cell Genomics: Approaches and Utility in Immunology. *Trends in immunology*, *38*(2), 140–149. https://doi.org/10.1016/j.it.2016.12.001
22. Bossel Ben-Moshe, N., Hen-Avivi, S., Levitin, N., Yehezkel, D., Oosting, M., Joosten, L., Netea, M. G., & Avraham, R. (2019). Predicting bacterial infection outcomes using single cell RNA-sequencing analysis of human immune cells. *Nature communications*, *10*(1), 3266. https://doi.org/10.1038/s41467-019-11257-y
23. Ren, X., Wen, W., Fan, X., Hou, W., Su, B., Cai, P., Li, J., Liu, Y., Tang, F., Zhang, F., Yang, Y., He, J., Ma, W., He, J., Wang, P., Cao, Q., Chen, F., Chen, Y., Cheng, X., Deng, G., … Zhang, Z. (2021). COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas. *Cell*, *184*(7), 1895–1913.e19. https://doi.org/10.1016/j.cell.2021.01.053
24. Laterre, P. F., François, B., Collienne, C., Hantson, P., Jeannet, R., Remy, K. E., & Hotchkiss, R. S. (2020). Association of Interleukin 7 Immunotherapy With Lymphocyte Counts Among Patients With Severe Coronavirus Disease 2019 (COVID-19). *JAMA network open*, *3*(7), e2016485. https://doi.org/10.1001/jamanetworkopen.2020.16485
25. Ren, Y., Shu, T., Wu, D., Mu, J., Wang, C., Huang, M., Han, Y., Zhang, X. Y., Zhou, W., Qiu, Y., & Zhou, X. (2020). The ORF3a protein of SARS-CoV-2 induces apoptosis in cells. *Cellular & molecular immunology*, *17*(8), 881–883. https://doi.org/10.1038/s41423-020-0485-9

26. Zhu, L., Yang, P., Zhao, Y., Zhuang, Z., Wang, Z., Song, R., Zhang, J., Liu, C., Gao, Q., Xu, Q., Wei, X., Sun, H. X., Ye, B., Wu, Y., Zhang, N., Lei, G., Yu, L., Yan, J., Diao, G., Meng, F., … Liu, W. J. (2020). Single-Cell Sequencing of Peripheral Mononuclear Cells Reveals Distinct Immune Response Landscapes of COVID-19 and Influenza Patients. *Immunity*, *53*(3), 685–696.e3. https://doi.org/10.1016/j.immuni.2020.07.009
27. Guo, Q., Zhao, Y., Li, J., Liu, J., Yang, X., Guo, X., Kuang, M., Xia, H., Zhang, Z., Cao, L., Luo, Y., Bao, L., Wang, X., Wei, X., Deng, W., Wang, N., Chen, L., Chen, J., Zhu, H., Gao, R., … You, F. (2021). Induction of alarmin S100A8/A9 mediates activation of aberrant neutrophils in the pathogenesis of COVID-19. *Cell host & microbe*, *29*(2), 222–235.e4. https://doi.org/10.1016/j.chom.2020.12.016
28. Paladino, L., Vitale, A. M., Caruso Bavisotto, C., Conway de Macario, E., Cappello, F., Macario, A., & Gammazza, A. M. (2020). The Role of Molecular Chaperones in Virus Infection and Implications for Understanding and Treating COVID-19. *Journal of clinical medicine*, *9*(11), 3518. https://doi.org/10.3390/jcm9113518
29. Lorente, L., Martín, M. M., Franco, A., Barrios, Y., Cáceres, J. J., Solé-Violán, J., Perez, A., Marcos Y Ramos, J. A., Ramos-Gómez, L., Ojeda, N., Jiménez, A., Working Group on COVID-19 Canary ICU, & Annex. Members of the BIOMEPOC group (2021). HLA genetic polymorphisms and prognosis of patients with COVID-19. Polimorfismos genéticos de los HLA y pronóstico de pacientes con COVID-19. *Medicina intensiva*, *45*(2), 96–103. https://doi.org/10.1016/j.medin.2020.08.004
30. Tavasolian, F., Rashidi, M., Hatam, G. R., Jeddi, M., Hosseini, A. Z., Mosawi, S. H., Abdollahi, E., & Inman, R. D. (2021). HLA, Immune Response, and Susceptibility to COVID-19. *Frontiers in immunology*, *11*, 601886. https://doi.org/10.3389/fimmu.2020.601886
31. Huerga Encabo, H., Grey, W., Garcia-Albornoz, M., Wood, H., Ulferts, R., Aramburu, I. V., Kulasekararaj, A. G., Mufti, G., Papayannopoulos, V., Beale, R., & Bonnet, D. (2021). Human Erythroid Progenitors Are Directly Infected by SARS-CoV-2: Implications for Emerging Erythropoiesis in Severe COVID-19 Patients. *Stem cell reports*, *16*(3), 428–436. https://doi.org/10.1016/j.stemcr.2021.02.001

Autoevaluación

Cuando escogí el tema de este TFG, lo hice porque había realizado este tipo de análisis durante mi estancia en prácticas y me parecía un tema interesante además de relativamente novedoso. Además, según las búsquedas que he hecho de empresas de cara a comenzar mi carrera laboral después de la universidad, vi que era algo que las empresas ofrecían como servicio o que en las ofertas de trabajo incluían como un 'plus'. Sin embargo, no pensé que con la realización del TFG fuera a meterme tanto en el tema e interesarme tanto. Durante las prácticas me limité a realizar el análisis, me limité a la parte más técnica, algo de interpretación de resultados, pero principalmente obtenía los resultados y ahí terminaba mi faena. Por eso este trabajo me ha gustado tanto realizarlo, porque he investigado para entender el porqué de cada paso, lo que me ha ayudado a realmente saber lo que estoy haciendo, y he aprendido a interpretar lo que genero con el análisis, he tenido que buscar la función de los genes y saber qué implica que se expresen o no, que lo hagan más o menos.

En cuanto al desarrollo del trabajo, me he encontrado con impedimentos, sobre todo tecnológicos. En diversas etapas del análisis he tenido problemas de capacidad de memoria, ya que lo he realizado con mi ordenador personal y sus recursos son limitados, por lo que he tenido que cambiar la forma de hacer algunas cosas o adaptar los datos para poder seguir adelante. Por eso una de las consideraciones que quiero dejar a cualquiera que desee realizar este tipo de análisis con grandes cantidades de datos es que tenga el hardware necesario para hacerlo ya que le facilitará mucho el trabajo.

Otro aspecto que considerar es que las funciones informáticas, pese a ser específicas para trabajar con este tipo de datos, no dejan de ser un conjunto de instrucciones que tratan los datos de manera general, por lo que en algunas ocasiones pueden aparecer resultados que no tengan mucho sentido, como que obtengas expresión de un gen en un determinado tipo celular y, a la hora de investigar sobre ese gen, descubras que no se expresa en el tipo celular que tú has obtenido. Esto te hace plantearte si has realizado bien el análisis, si hay algún paso que no hayas ejecutado de manera correcta, pero también hay que entender que, como he dicho sobre estas funciones, no están hechas para realizar justo el análisis que estás haciendo, por lo que pueden fallar o dar resultados a priori incongruentes. Siempre que estos fallos no sean demasiado relevantes, o no sean demasiado comunes, es tarea del investigador filtrar y discutir los resultados para ofrecer la visión más precisa de lo que se obtiene experimentalmente, y personalmente creo que eso he hecho en este trabajo.

En conclusión, considero que la realización de este trabajo ha sido muy positiva para mí, aportándome conocimientos y experiencia por partes iguales, y estoy contento por haber realizado un TFG que combine los ámbitos del doble grado que estoy terminando, la biotecnología y la informática, ya que para eso me metí en esta carrera.

Anexos

En el apartado de anexos incluyo el código del script que he elaborado para realizar el análisis de expresión diferencial.

Anexo 1: Script de análisis de expresión diferencial

```
##############################################################################################

# Script para la realización del analisis de expresión diferencial correspondiente al TFG utilizando las #

# funciones de la libreria Seurat del grupo SatijaLab y programado en lenguaje R. #

# Autor: Marcos Esteve Hernández #

##############################################################################################

# Fijar directorio de trabajo y cargar librerias utilizadas

setwd("C:/Users/34659/OneDrive/Escritorio/TFG Biotec/Differential Expression Test")

library(dplyr)

library(Seurat)

library(patchwork)

library(ggplot2)

library(cowplot)

library(readr)

library(writexl)


covid1 <- readRDS(file = "./Data/GSM4557327_555_1_cell.counts.matrices.rds")

covid1 <- CreateSeuratObject(counts = covid1$exon, project = "covid1", min.cells = 33, min.features = 200)

covid2 <- readRDS(file = "./Data/GSM4557328_555_2_cell.counts.matrices.rds")

covid2 <- CreateSeuratObject(counts = covid2$exon, project = "covid2", min.cells = 33, min.features = 200)

covid3 <- readRDS(file = "./Data/GSM4557329_556_cell.counts.matrices.rds")

covid3 <- CreateSeuratObject(counts = covid3$exon, project = "covid3", min.cells = 33, min.features = 200)

covid4 <- readRDS(file = "./Data/GSM4557330_557_cell.counts.matrices.rds")

covid4 <- CreateSeuratObject(counts = covid4$exon, project = "covid4", min.cells = 33, min.features = 200)

covid5 <- readRDS(file = "./Data/GSM4557331_558_cell.counts.matrices.rds")

covid5 <- CreateSeuratObject(counts = covid5$exon, project = "covid5", min.cells = 33, min.features = 200)

covid6 <- readRDS(file = "./Data/GSM4557332_559_cell.counts.matrices.rds")

covid6 <- CreateSeuratObject(counts = covid6$exon, project = "covid6", min.cells = 33, min.features = 200)

covid7 <- readRDS(file = "./Data/GSM4557333_561_cell.counts.matrices.rds")
```

```
covid7 <- CreateSeuratObject(counts = covid7$exon, project = "covid7", min.cells = 33, min.features = 200)

control1 <- readRDS(file = "./Data/GSM4557334_HIP002_cell.counts.matrices.rds")

control1 <- CreateSeuratObject(counts = control1$exon, project = "control1", min.cells = 33, min.features = 200)

control2 <- readRDS(file = "./Data/GSM4557335_HIP015_cell.counts.matrices.rds")

control2 <- CreateSeuratObject(counts = control2$exon, project = "control2", min.cells = 33, min.features = 200)

control3 <- readRDS(file = "./Data/GSM4557336_HIP023_cell.counts.matrices.rds")

control3 <- CreateSeuratObject(counts = control3$exon, project = "control3", min.cells = 33, min.features = 200)

control4 <- readRDS(file = "./Data/GSM4557337_HIP043_cell.counts.matrices.rds")

control4 <- CreateSeuratObject(counts = control4$exon, project = "control4", min.cells = 33, min.features = 200)

control5 <- readRDS(file = "./Data/GSM4557338_HIP044_cell.counts.matrices.rds")

control5 <- CreateSeuratObject(counts = control5$exon, project = "control5", min.cells = 33, min.features = 200)

control6 <- readRDS(file = "./Data/GSM4557339_HIP045_cell.counts.matrices.rds")

control6 <- CreateSeuratObject(counts = control6$exon, project = "control6", min.cells = 33, min.features = 200)

#Integrar todos los Seurat Object en uno solo

covid_full <- merge(covid1, y = c(covid2, covid3, covid4, covid5, covid6, covid7,control1, control2, control3, control4, control5, control6), add.cell.ids = c('covid1', 'covid2', 'covid3', 'covid4', 'covid5', 'covid6', 'covid7', 'control1', 'control2', 'control3', 'control4', 'control5', 'control6'), project = 'covid_project')

# Borrar variables que ya no se utilizarán

rm(covid1, covid2, covid3, covid4, covid5, covid6, covid7, control1, control2, control3, control4, control5, control6)

# Guardado de seguridad

#saveRDS(covid_full, file = "./covid_merged.rds")

#covid_full <- readRDS(file = "./covid_merged.rds")

# Control de calidad de las muestras

covid_full[["percent.mt"]] <- PercentageFeatureSet(covid_full, pattern = "^MT-")

# Visualizar las metricas de control de calidad

VlnPlot(covid_full, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)

covid_full <- subset(covid_full, subset = nFeature_RNA > 200 & nFeature_RNA < 2000 & percent.mt < 5)

# Separar el objeto por su identificador

covid_splited <- SplitObject(covid_full, split.by = "ident")

# Normalizar los datos

covid_splited <- lapply(X = covid_splited, FUN = function(x) {

  x <- NormalizeData(x)

  x <- FindVariableFeatures(x, selection.method = "vst", nfeatures = 2000)
```

```
})

rm(covid_full)

# Gráfico de VariableFeatures

top10 <- head(VariableFeatures(covid_splited$covid4), 10)

plot1 <-VariableFeaturePlot(covid_splited$covid4)

plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)

plot1 + plot2

rm(top10, plot1, plot2)

# Separar el conjunto para poder integrar

covid_splited1 <- c(covid1=covid_splited[["covid1"]],covid2=covid_splited[["covid2"]],covid3=covid_splited[["covid3"]],covid4=covid_splited[["covid4"]],covid5=covid_splited[["covid5"]],covid6=covid_splited[["covid6"]],covid7=covid_splited[["covid7"]])

covid_splited2 <- c(control1=covid_splited[["control1"]],control2=covid_splited[["control2"]],control3=covid_splited[["control3"]],control4=covid_splited[["control4"]],control5=covid_splited[["control5"]],control6=covid_splited[["control6"]])

saveRDS(covid_splited1, file = "./covid_splited1.rds")

covid_splited1 <- readRDS(file = "./covid_splited1.rds")

saveRDS(covid_splited2, file = "./covid_splited2.rds")

covid_splited2 <- readRDS(file = "./covid_splited2.rds")

# Data reintegration

covid.anchors <- FindIntegrationAnchors(object.list = covid_splited, dims = 1:20)

covid_combined <- IntegrateData(anchorset = covid.anchors, dims = 1:20)

rm(covid_splited, covid.anchors)

# Crear una columna que agrupe por enfermedad

samp <- covid_combined[["orig.ident"]]

rep<-sample(1, nrow(covid_combined[["orig.ident"]]), replace = TRUE)

names<-c("covid", "healthy")

for (i in 1:length(rep)) {

        if(samp[i,1]==samp[1,1] | samp[i,1]==samp[3765,1] | samp[i,1]==samp[8680,1] | samp[i,1]==samp[9720,1] | samp[i,1]==samp[14400,1] | samp[i,1]==samp[17255,1] | samp[i,1]==samp[18300,1]){rep[i]<-1}

        else if(samp[i,1]==samp[19760,1] | samp[i,1]==samp[20370,1] | samp[i,1]==samp[20680,1] | samp[i,1]==samp[21125,1] | samp[i,1]==samp[22175,1] | samp[i,1]==samp[23600,1]){rep[i]<-2}

        }

my_factor<-factor(rep, labels=names)
```

```
covid_combined[["orig.ident"]]<-my_factor

rm(my_factor, rep, samp, names, i)

# Clusterizacion

covid_combined <- ScaleData(covid_combined, verbose = FALSE)

covid_combined <- RunPCA(covid_combined, verbose = FALSE)

# Determinar la dimensionalidad

#jack <- JackStraw(covid_combined, num.replicate = 100)

#jack <- ScoreJackStraw(covid_combined, dims = 1:20)

#JackStrawPlot(jack, dims = 1:20)

ElbowPlot(covid_combined)

covid_combined <- RunUMAP(covid_combined, reduction = "pca", dims = 1:15)

covid_combined <- FindNeighbors(covid_combined, reduction = "pca", dims = 1:15)

covid_combined <- FindClusters(covid_combined, resolution = 0.5)

#DimPlot(covid_combined, reduction = "umap", label = TRUE)

DefaultAssay(covid_combined) <- "RNA"

# Encontrar los marcadores de cada cluster independientemente de la condicion para identificar los tipos celulares presentes

c0.markers <- FindConservedMarkers(covid_combined, ident.1 = 0, grouping.var = "orig.ident", verbose = FALSE)

c1.markers <- FindConservedMarkers(covid_combined, ident.1 = 1, grouping.var = "orig.ident", verbose = FALSE)

c2.markers <- FindConservedMarkers(covid_combined, ident.1 = 2, grouping.var = "orig.ident", verbose = FALSE)

c3.markers <- FindConservedMarkers(covid_combined, ident.1 = 3, grouping.var = "orig.ident", verbose = FALSE)

c4.markers <- FindConservedMarkers(covid_combined, ident.1 = 4, grouping.var = "orig.ident", verbose = FALSE)

c5.markers <- FindConservedMarkers(covid_combined, ident.1 = 5, grouping.var = "orig.ident", verbose = FALSE)

c6.markers <- FindConservedMarkers(covid_combined, ident.1 = 6, grouping.var = "orig.ident", verbose = FALSE)

c7.markers <- FindConservedMarkers(covid_combined, ident.1 = 7, grouping.var = "orig.ident", verbose = FALSE)

c8.markers <- FindConservedMarkers(covid_combined, ident.1 = 8, grouping.var = "orig.ident", verbose = FALSE)

c9.markers <- FindConservedMarkers(covid_combined, ident.1 = 9, grouping.var = "orig.ident", verbose = FALSE)

c10.markers <- FindConservedMarkers(covid_combined, ident.1 = 10, grouping.var = "orig.ident", verbose = FALSE)

c11.markers <- FindConservedMarkers(covid_combined, ident.1 = 11, grouping.var = "orig.ident", verbose = FALSE)

c12.markers <- FindConservedMarkers(covid_combined, ident.1 = 12, grouping.var = "orig.ident", verbose = FALSE)

c13.markers <- FindConservedMarkers(covid_combined, ident.1 = 13, grouping.var = "orig.ident", verbose = FALSE)

c14.markers <- FindConservedMarkers(covid_combined, ident.1 = 14, grouping.var = "orig.ident", verbose = FALSE)

c15.markers <- FindConservedMarkers(covid_combined, ident.1 = 15, grouping.var = "orig.ident", verbose = FALSE)
```

```
c16.markers <- FindConservedMarkers(covid_combined, ident.1 = 16, grouping.var = "orig.ident", verbose = FALSE)

c17.markers <- FindConservedMarkers(covid_combined, ident.1 = 17, grouping.var = "orig.ident", verbose = FALSE)

c18.markers <- FindConservedMarkers(covid_combined, ident.1 = 18, grouping.var = "orig.ident", verbose = FALSE)

c19.markers <- FindConservedMarkers(covid_combined, ident.1 = 19, grouping.var = "orig.ident", verbose = FALSE)

## La identificación se ha realizado mediante la búsqueda de la presencia de los marcadores en la base de datos de proteinatlas.org ##

# Renombramos los clústeres con los tipos celulares identificados

covid_combined <- RenameIdents(covid_combined, '0' = "CD8 T", '1' = "Monocito Clasico", '2' = "Naive CD4 T", '3' = "Monocito Intermedio", '4' = "Natural Killer", '5' = "No concluyente", '6' = "CD4 T", '7' = "Monocito Clasico", '8' = "Memory B", '9' = "Naive B", '10' = "Neutrofilo", '11' = "Monocito No Clasico", '12' = "Basofilo", '13' = "Regulador T", '14' = "Eritroblasto", '15' = "Basofilo", '16' = "DC")

# Obtener los genes con diferencia de expresión entre condiciones

covid_combined$celltype.dis <- paste(Idents(covid_combined), covid_combined$orig.ident, sep = " - ")

covid_combined$celltype <- Idents(covid_combined)

Idents(covid_combined) <- "celltype.dis"

# CD8 T

covid_healthy <- FindMarkers(covid_combined, ident.1 = "CD8 T - covid", ident.2 = "CD8 T - healthy", verbose = FALSE)

a_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Monocito Clasico

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Monocito Clasico - covid", ident.2 = "Monocito Clasico - healthy", verbose = FALSE)

b_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Naive CD4 T

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Naive CD4 T - covid", ident.2 = "Naive CD4 T - healthy", verbose = FALSE)

c_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Monocito Intermedio

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Monocito Intermedio - covid", ident.2 = "Monocito Intermedio - healthy", verbose = FALSE)

d_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Natural killer

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Natural Killer - covid", ident.2 = "Natural Killer - healthy", verbose = FALSE)

e_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# CD4 T
```

```
covid_healthy <- FindMarkers(covid_combined, ident.1 = "CD4 T - covid", ident.2 = "CD4 T - healthy", verbose = FALSE)

f_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Memory B

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Memory B - covid", ident.2 = "Memory B - healthy", verbose = FALSE)

g_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Naive B

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Naive B - covid", ident.2 = "Naive B - healthy", verbose = FALSE)

h_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Neutrofilo

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Neutrofilo - covid", ident.2 = "Neutrofilo - healthy", verbose = FALSE)

i_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Monocito No Clasico

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Monocito No Clasico - covid", ident.2 = "Monocito No Clasico - healthy", verbose = FALSE)

j_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Basofilo

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Basofilo - covid", ident.2 = "Basofilo - healthy", verbose = FALSE)

k_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Regulador T

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Regulador T - covid", ident.2 = "Regulador T - healthy", verbose = FALSE)

l_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

# Eritroblasto

covid_healthy <- FindMarkers(covid_combined, ident.1 = "Eritroblasto - covid", ident.2 = "Eritroblasto - healthy", verbose = FALSE)

m_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

#Dendritic cell

covid_healthy <- FindMarkers(covid_combined, ident.1 = "DC - covid", ident.2 = "DC - healthy", verbose = FALSE)

n_data <- data.frame("gene"=row.names(covid_healthy), covid_healthy)

rm(covid_healthy)

#Información sobre el análisis
```

```
Information <-array(c("The results data frame has the following columns :",

                      "p_val : p_value (unadjusted)",

                      "avg_logFC : log fold-change of the average expression between the two groups. Positive values indicate that the feature is more highly expressed in the mild CoVid group.",

                     "pct.1 : The percentage of cells where the feature is detected in the CoVid group",

                     "pct.2 : The percentage of cells where the feature is detected in the healthy group",

                      "p_val_adj : Adjusted p-value, based on bonferroni correction using all features in the dataset.*",

                      "* p_val_adj is corrected by the number of genes used in the analysis with the goal of avoiding false positive occurrences. Note that it might generate many false negatives.",

                      "References:",

                      "https://satijalab.org/seurat/v3.0/de_vignette.html",
"http://www.biostathandbook.com/multiplecomparisons.html#:~:text=The%20Bonferroni%20correction%20is%20appropriate,two%20that%20might%20be%20significant."))

Information<-data.frame(Information)

# Guardamos los resultados en un excel

mild_covid.vs.healthy <- list("CD8 T"=a_data,"Monocito Clasico"=b_data, "Naive CD4 T"=c_data, "Monocito Intermedio"=d_data, "NK"=e_data, "CD4 T"=f_data, "Memory B"=g_data, "Naive B"=h_data, "Neutrofilo"=i_data, "Monocito No Clasico"=j_data, "Basofilo"=k_data, "Regulador T"=l_data, "Eritroblasto"=m_data, "DC"=n_data, "Information"=Information)

write_xlsx(mild_covid.vs.healthy,"./Covid_vs_healthy.xlsx")
```