Automatización de pipelines de secuenciación de RNA en sistemas distribuidos

Trabajo fin de grado de Ingeniería Informática

Área de arquitectura de computadores y sistemas operativos

Autor: Iago Pinal Fernández

Consultora: Belen Bermejo González

Profesor responsable de la asignatura: Josep Jorba Esteve

Para Ruth

"En medio de toda dificultad se encuentra una oportunidad"

Albert Einstein

Índice

Resumen

Automatizar y estandarizar el procesamiento de datos de secuenciación de RNA es esencial para garantizar la calidad de los resultados de este tipo de análisis. Además, esto permite una mayor eficiencia en el análisis y la interpretación de los datos, ya que se pueden aplicar automáticamente métodos estandarizados para la limpieza y el preprocesamiento de los datos. Por otra parte poder realizar análisis exploratorios de manera ágil es crítico para poder diseñar estudios confirmatorios con rapidez.

En este proyecto, se diseñó un sistema escalable para automatizar pipelines de secuenciación de RNA en sistemas distribuidos y se desarrolló un software de visualización de este tipo de datos para poder realizar análisis exploratorios de manera eficiente.

Para demostrar la utilidad de estas herramientas se adjuntan tres estudios científicos que se realizaron en los últimos meses utilizando el producto de este trabajo fin de carrera.

Palabras clave: Automatizacion, sistemas distribuidos, secuenciacion de RNA.

Introducción

Las miopatías inflamatorias son enfermedades autoinmunes sistémicas, como el lupus o la esclerodermia, en las que una respuesta inmunológica alterada contra lo propio genera daño en múltiples órganos y tejidos, incluyendo el músculo, pero también la piel, los pulmones o las articulaciones. Los grupos clínicos más universalmente aceptados son la dermatomiositis, el síndrome por anticuerpos antisintetasa, la miopatía necrosante inmunomediada y la miopatía por cuerpos de inclusión.(1)

Uno de los avances más relevantes para la compresión de estas enfermedades ha sido el descubrimiento de autoanticuerpos en el suero de los pacientes. En todos los grupos clínicos menos en la miopatía por cuerpos de inclusión se han identificado autoanticuerpos específicos que generalmente son mutuamente excluyentes y definen fenotipos clínicos muy homogéneos.(1) Tanto es así que se ha postulado que estos autoanticuerpos pueden estar definiendo enfermedades distintas.(2)

El segundo avance más importante en el campo de las miopatías inflamatorias fue la utilización de estudios de transcriptómica para entender que vías inflamatorias están alteradas en cada tipo de síndrome clínico. Por ejemplo, estos estudios descubrieron la importancia de la vía del interferón de tipo I en la dermatomiositis, lo que ha modificado significativamente el manejo clínico de estos pacientes.(3,4)

Nuestra investigación se centra en entender las características clínicas y la patogénesis de cada grupo clínico y serológico de miositis. Para esto, hemos estado utilizando una serie

de técnicas transcriptómicas basadas en la secuenciación de RNA de músculo de pacientes con miopatía inflamatoria. Esto nos ha llevado a descubrir marcadores transcriptómicos y vías inflamatorias específicas de cada tipo de miositis, lo que ha probado ser un excelente generador de hipótesis para estudios confirmatorios posteriores.(3,5–9)

La forma más básica de secuenciación de RNA es secuenciar el RNA total de un tejido, en este caso músculo humano. Para esto, primero se tiene que extraer el RNA, lo que es complicado en el caso de tejidos poco friables como el músculo, ya que hay que buscar un buen balance entre la eficiencia de extracción de RNA y el nivel de degradación de dicha molécula. Una vez aislado, se tiene que disminuir la cantidad de RNA ribosomal, lo generalmente se logra bien aislando el RNA maduro con cola poli-A o fragmentando el RNA ribosomal de manera dirigida. Tras haber aislado el RNA ribosomal se genera la librería de RNA que incluye los adaptadores que hacen que el RNA se pueda unir al soporte de secuenciación y los índices que permiten que múltiples muestras se puedan secuenciar simultáneamente.(5)

Una vez secuenciadas las muestras se comienza el procesamiento bioinformático de las mismas. El primer archivo que se obtiene de la secuenciación es un archivo denominado *binary base call* o bcl que incluye varios experimentos simultáneos. El primer paso del análisis es separar los experimentos individuales incluidos en el archivo bcl, proceso que se denomina demultiplexación. Para realizar este paso, ademas del archivo bcl se requieren los índices de cada una de las muestras y sus identificadores. El software más popular para realizar este paso es bcl2fastq. Este paso, que teóricamente el más trivial del proceso de análisis, no está libre de problemas, un defecto en la indexación de las secuencia de RNA puede generar índices arbitrarios, lo que puede llevar a contaminación del resto de muestras del *batch*. La presencia de lecturas de genes que no suelen expresarse en el músculo en cantidad relativamente estable sugiere un artefacto de este tipo.

Tras demutiplexar el archivo bcl se obtienen archivos .fastq para cada una de las muestras en cuestión. Los archivos .fastq tienen secuencias de cuatro filas como la que se muestra

a continuación en las que se incluyen los metadatos de cada secuencia (línea 1), su código de nucleótidos (línea 2) y la calidad de secuenciación de cada nucleótido, codificada con un código ASCII (línea 4). Estos archivos .fastq se pueden comprimir para ahorrar espacio en disco.(10)

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Durante el proceso de secuenciación se pueden generar varios artefactos técnicos, como la contaminación con adaptadores o la presencia de secuencias con baja calidad, que se pueden solucionar de manera determinista. En cuanto a la contaminación con adaptadores, se pueden eliminar aquellos nucleótidos que tengan una secuencia identificada como correspondiente a uno de los reactivos utilizados en la secuenciación. Para solucionar secuencias o fragmentos de secuencia con baja calidad, se puede elegir un punto de corte de calidad y eliminar aquellos nucleótidos por debajo de este punto de corte. Programas populares para realizar este paso con Trimmomatic o fastp,(11,12) teniendo este último como ventaja que aplica simultáneamente el control de calidad del archivo .fastq resultante.

Una vez se tiene el archivo .fastq limpio de artefactos técnicos, el siguiente paso es alinear cada una de las secuencias al genoma de referencia para identificar a que región del mismo corresponde. Este es un punto crítico del proceso y es uno de los más complicados técnicamente dado que el RNA, a diferencia del DNA, sufre un proceso de splicing, en el que las regiones de secuencia denominadas intrones se eliminan. Por tanto, las secuencias de RNA están compuestas por secciones lineales discontinuas de longitud variable correspondientes al genoma de referencia. Para hacer el proceso más complicado, cada gen puede tener múltiples variantes de splicing, generando diferentes tránscritos del mismo gen y cada individuo tiene mutaciones que lo diferencian de la secuencia de referencia. Además, hay regiones repetidas del genoma en que una misma secuencia puede corresponderse con regiones diferentes y hay

regiones del genoma que varían considerablemente, como la región del HLA o de la cadena variable de immunoglobulinas, lo que dificulta su alineamiento.

Hay diversas técnicas para disminuir la incertidumbre en la secuenciación, como aumentar la longitud del fragmento secuenciado, o secuenciar los dos extremos de la hebra de RNA (*paired-end*), lo que hace que esos dos fragmentos tengan que estar cerca del uno del otro en genoma de referencia, permitiendo disminuir las dudas en la asignación durante el alineamiento.

Existen múltiples alineadores disponibles para realizar este paso del proceso, con algoritmos diferentes y consumo de recursos variable. Este es el paso del proceso con mayores requerimientos computacionales y, además de la fiabilidad en la asignación de las secuencias es crítica la velocidad en el procesamiento de las mismas y la cantidad de recursos computacionales necesarios. Dos de las herramientas más populares que existen son el alineador STAR y Salmon.(13,14) STAR es muy fiable pero relativamente lento y con unos requerimientos de memoria RAM elevados, llegando a los 45Gb para genomas humanos.(13) En cuanto a Salmon, utiliza una estrategia de pseudoalineamiento que le confiere una velocidad mucho mayor con un nivel de fiabilidad aceptable.(14) A diferencia de STAR, que genera archivos que incluyen el alineamiento nucleótido por nucleótido de las secuencias (archivos .sam o, su versión comprimida .bam), Salmon realiza directamente la cuantificación de los tránscritos, lo que lo hace inválido para aplicaciones en las que se necesite saber con exactitud el alineamiento de cada nucleótido.

Algunos alineadores (como STAR), no requieren necesariamente un genoma de referencia para realizar el alineamiento y pueden utilizar las propias secuencias de RNA para deducir la contigüidad de las secuencias. No obstante, en genomas bien estudiados, como el humano, para la mayoría de aplicaciones se utiliza un genoma y unos tránscritos de referencia para facilitar el análisis, aumentar la velocidad y priorizar la cantidad de información resultante. Actualmente, para los organismos más estudiados hay varios genomas de referencia

mantenidos por organizaciones científicas diferentes. Estos genomas de referencia son en muchos aspectos similares pero no son iguales, y para ciertas aplicaciones esto puede afectar a los resultados obtenidos (15,16). Además, el formato de estos genomas de referencia no es exactamente igual, lo que genera problemas al intentar utilizar las diferentes herramientas bioinformáticas con diferentes archivos. Los dos genomas de referencia más populares son RefSeq, mantenido por el NCBI en Estados Unidos, y GENCODE/Ensembl, mantenido por un consorcio de investigación pública. Los transcritos de GENCODE/Ensemble cubren más regiones genómicas, y capturan más variantes, lo que lo hace atractivo para proyectos más exploratorios (15).

Tras realizar este paso, o bien se obtienen archivos de alineamiento (.sam o .bam) o bien se obtiene directamente el número de cuentas de cada tránscrito. En el primero de los casos se tiene que calcular los diferentes tránscritos en base al archivo de alineamiento para lograr el número de cuentas de cada tránscrito. Una vez obtenido el número de cuentas de cata tránscrito se tienen sumar el número de cuentas de todos los tránscritos de cada gen para poder así obtener el archivo de cuentas crudo, que es una matriz de datos en que cada columna se corresponde con una muestra y cada fila con un gen. Para estudios en los que el interés es estudiar los diferentes tránscritos de cada gen obviamente no será preciso realizar la suma de los tránscritos.

Una vez obtenido el archivo de cuentas habitualmente el siguiente paso es normalizar dichas cuentas para eliminar sesgos sistemáticos en el número de cuentas total de experimento o la longitud del gen. Un experimento con el doble de número de cuentas es esperable que tenga el doble de cuentas de cada gen, asimismo, un gen el doble de largo que otro es de esperar que tenga el doble de cuentas. Así, las formas más sencillas de normalización realizan simplemente una división entre los millones de cuentas del experimento y luego entre el número de kilobases del gen (framentos por millon de kilobases o FPKM) o bien al reves, dividiendo el número de kilobases del gen y luego el número de millones de cuentas del

experimento (transcritos por kilobase millón o TPM). Entre FPKM y TPM el TPM tiene la ventaja de que el número de cuentas total es siempre el mismo. No obstante, estos sistemas de normalización simplistas ignoran ciertos aspectos técnicos de la secuenciación de RNA, como por ejemplo que no hay la misma profundidad de secuenciación en secuencias cortas que en secuencias largas. Para esto se han desarrollado algoritmos más sofisticados que tienen en cuenta la distribución de las cuentas para cada grupo de muestras y diferentes poblaciones de RNA. Así, por ejemplo, paquetes de software populares para el cálculo de la expresión diferencial como DESEq2 o limma-voom,(17,18) utilizan estos sistemas. En concreto limma-voom genera un estadístico de cuentas normalizadas muy popular denominado *trimmed mean of M-values* o TMM procedente del paquete EdgeR.(18,19)

El análisis de expresión diferencial es, a menudo, uno de los primeros estudios de interés en este tipo de experimentos. El objetivo es determinar el número de veces que un tránscrito está sobre o infra expresado en una condición comparada con la otra. A menudo este tipo de análisis tienen que tener en cuenta variables confusoras o realizarse sobre subgrupos de muestras. Hay varios tipos de aproximaciones al estudio de expresión diferencial desde el punto de vista de los datos de inicio. Un grupo muy popular de programas utilizan valores de cuentas normalizados, esto tiene la ventaja de que se pueden utilizar si sólo se dispone de este tipo de datos (por ejemplo en base de datos públicas), no obstante ignoran la distribución de las poblaciones de RNA y en general tienen un peor comportamiento para genes con expresión baja. Otra alternativa es usar el número de cuentas crudo y normalizar los datos de manera interna teniendo en cuenta la distribución del número de cuentas de RNAs para realizar la expresión diferencial. Esta es la aproximación que usa DESEq2 o limma-voom y, en nuestra experiencia, produce resultados más robustos.(18,19)

Las cuentas normalizadas que mencionamos previamente se utilizan también para el análisis gráfico de los datos. No obstante, la producción celular de RNA es exponencial y es habitual hacer una transformación logarítmica ($\log_2$[cuentas-normalizadas+1]) antes de utilizarlas.

Dicho análisis gráfico tiene una serie de figuras que se usan frecuentemente y, por tanto, se puede automatizar hasta cierto grado. En primer lugar, es habitual tener que mostrar la expresión de un gen o un grupo de genes en diferentes grupos de interés. Para esto se pueden utilizar diagramas de cajas y, si se quiere añadir la información de cada muestra individual, solapar sobre el diagrama de cajas un *swarmplot* (Figura 1). Si se quiere mostrar la información de múltiples genes en varios grupos de una manera más compacta se puede utilizar sólo el diagrama de cajas, o aún más eficiente en términos espaciales, mostrar el intervalo de confianza al 95% de la distribución (Figura 2).

Otra aplicación común es necesitar mostrar el nivel de expresión de múltiples genes en todas las muestras o en varias condiciones. Este tipo de matrices de datos se pueden visualizar satisfactoriamente realizando el Z-score ($[x - \mu]/\omega$) y traduciéndolo a una escala de colores centrada en el 0 (Figura 2). Alternativamente también se obtienen buenos resultados usando un escalado min-max (Figura 3). Cuando el objetivo es mostrar la diferencia entre dos grupos, puede ser informativo usar gráficos mostrando el logaritmo del *fold-change* (Figura 4).

Un análisis gráfico muy habitual es querer visualizar la correlación entre dos genes, bien para un número bajo de genes o para números más elevados. Los gráficos de puntos, con líneas de regresión si es preciso, son óptimos para mostrar la correlación entre listas pequeñas de genes (Figura 6). Alternativamente, para comparaciones bivariadas más grandes se puede mostrar la correlación de Spearman codificada como un *heatmap* (Figura 7).

En ocasiones es necesario mostrar la estructura relacional de series de observaciones. Esto puede ser de interés con *bulk* RNAseq para identificar la estructura de una determinada condición o detectar problemas técnicos con determinadas muestras. Alternativamente en nuevas técnicas de secuenciación de RNA de célula única, esto es preciso para identificar los distintos tipo de células a analizar. Existen diferentes algoritmos para realizar este tipo de análisis, incluyendo el *principal component analysis* (PCA), el *t-distributed stochastic neigh-*

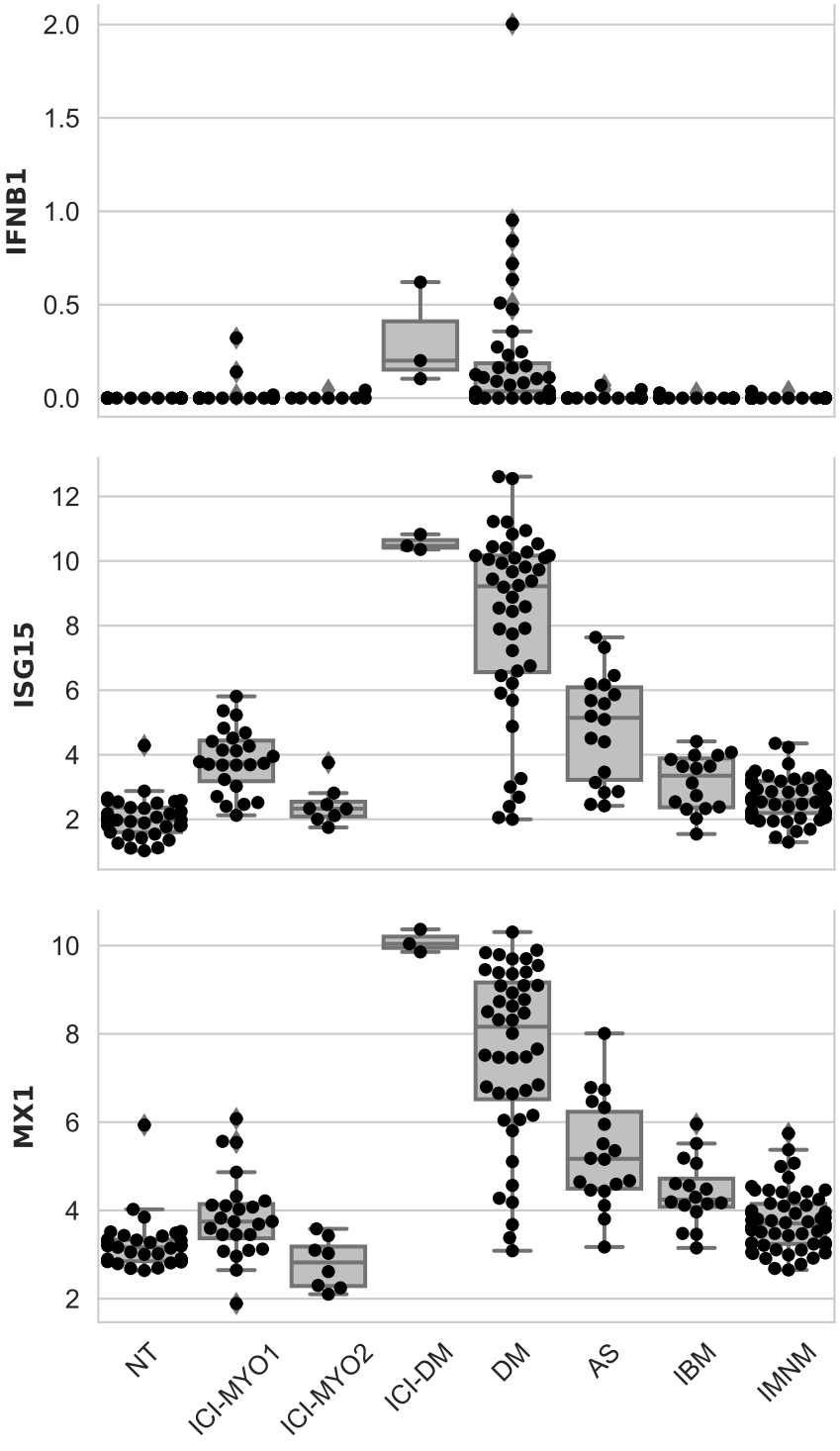


Figura 1. Ejemplo de swarmplot con diagrama de cajas

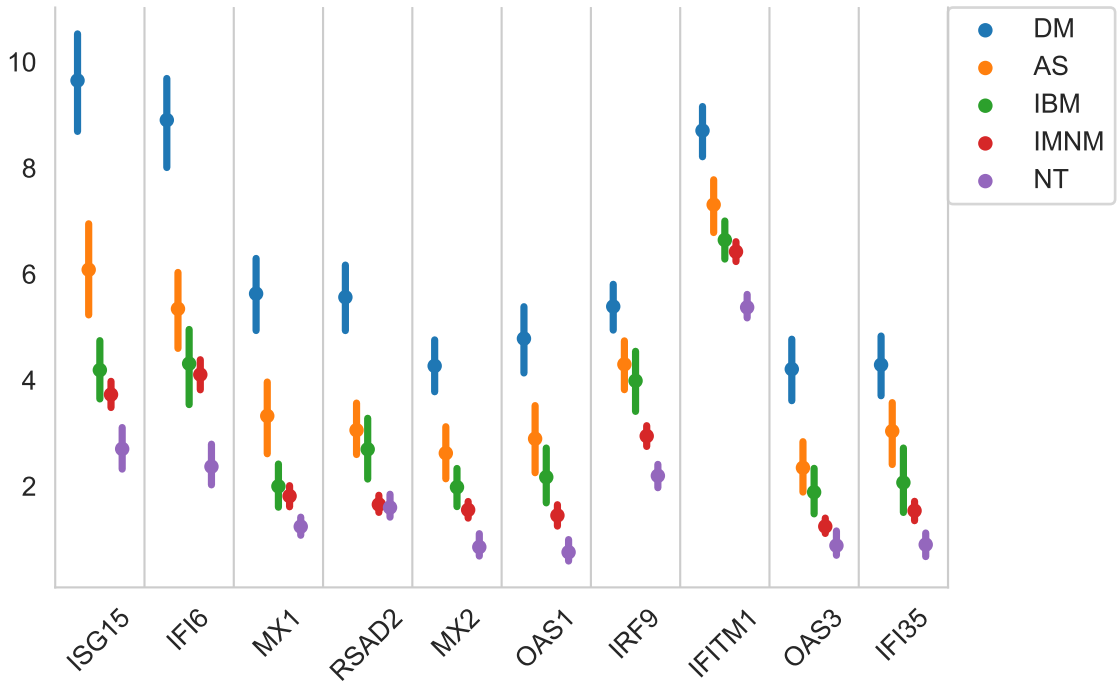


Figura 2. Ejemplo de gráfico con intervalos de confianza al 95%

bor embedding (t-SNE), o el *uniform manifold approximation and projection for dimension reduction* (UMAP) (Figura 8-9). Para estudios de secuenciación de RNA de célula única el UMAP tiene un mejor comportamiento y se utiliza en detrimento del t-SNE.

Una vez analizada la expresión diferencial y el análisis gráfico, uno de los análisis frecuentemente necesarios en este campo es la realización de análisis de *pathways*. Este tipo de análisis consiste en identificar pathways que estén implicados en las diferencias que se observan entre dos grupo. Para hacer este análisis existen dos técnicas fundamentales, el análisis de sobrerrepresentación y las aproximaciones basadas en *functional class scoring*.(20) El primer grupo de técnicas utiliza métodos de estadística inferencial basadas en la abundancia de genes presentes en cada lista. Este tipo de aproximación tiene la limitación fundamental de que ignora la magnitud de la sobre o infraexpresión de cada gen, lo que supone una pérdida de información significativa. Alternativamente, las aproximaciones basadas en *functional class scoring*, como el *Gene Set Enrichment Analysis* (GSEA),(21) consisten en ordenar los genes en base a un score y estudiar la distribución de los genes de cada *pathway* asumiendo

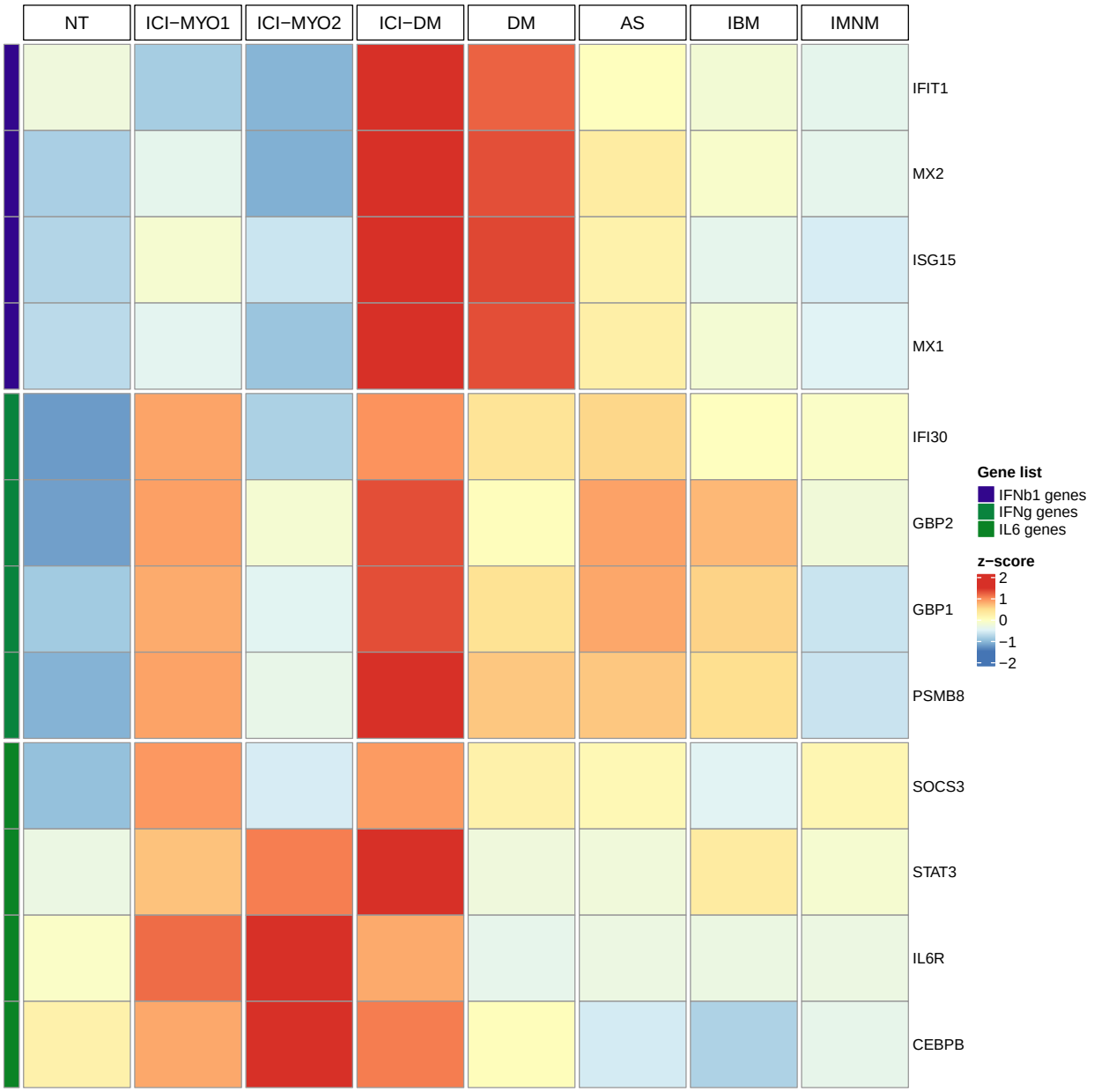


Figura 3. Ejemplo de gráfico de Z-scores

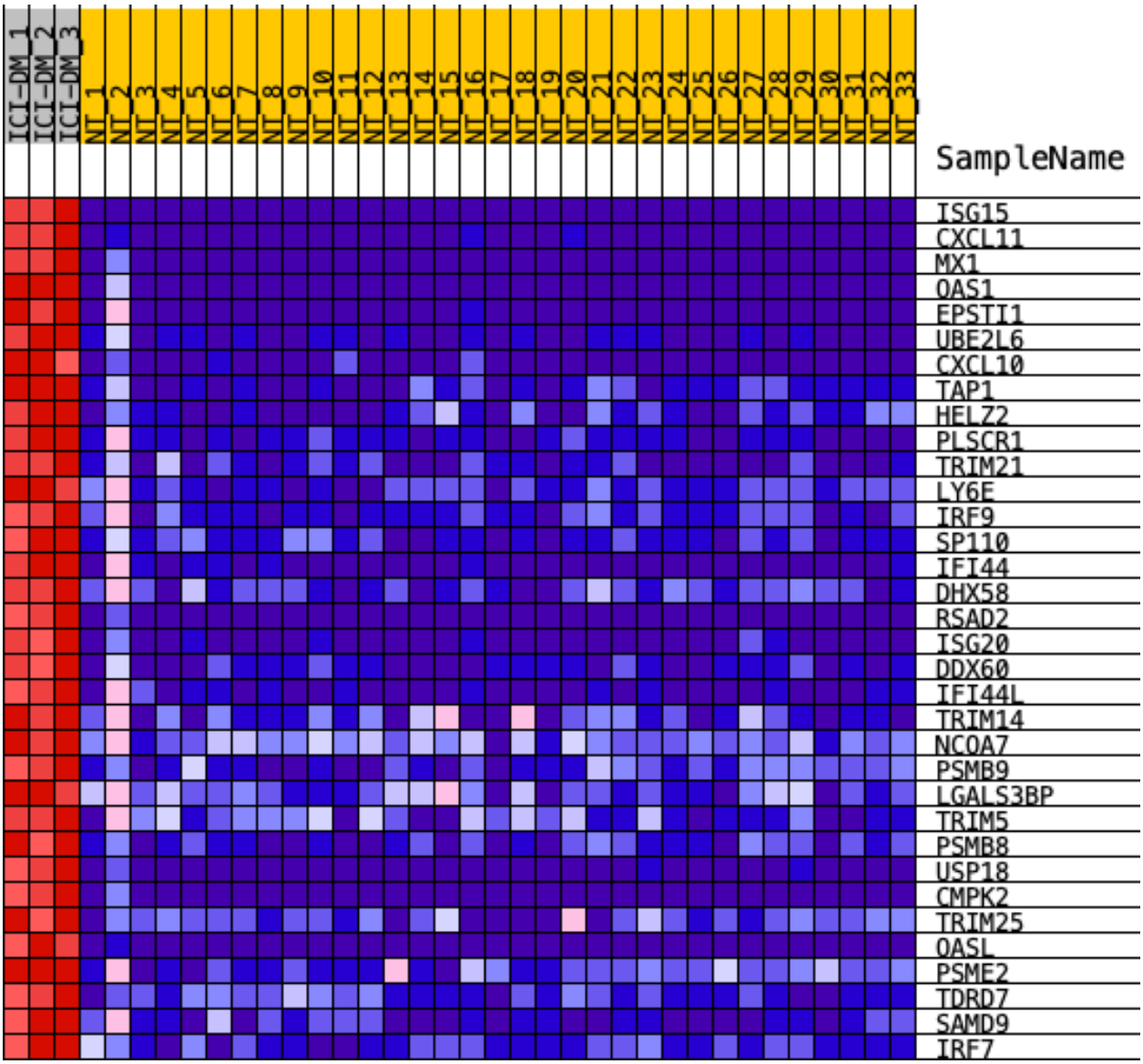


Figura 4. Ejemplo de gráfico con escalado min-max

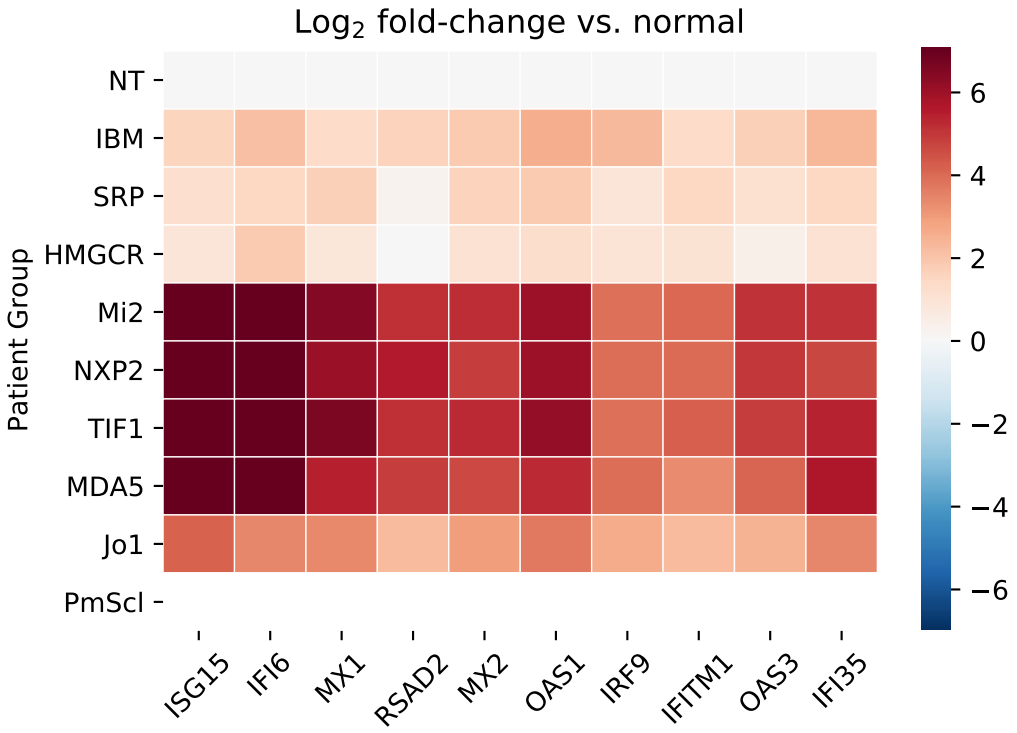


Figura 5. Ejemplo de gráfico de *fold-change*

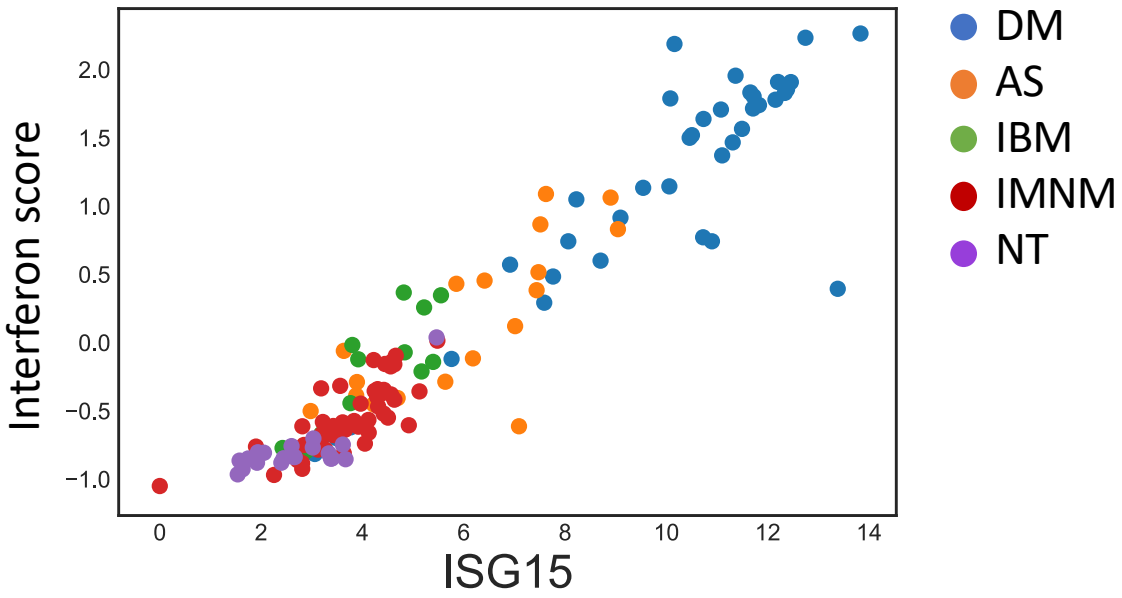


Figura 6. Ejemplo de gráfico de puntos

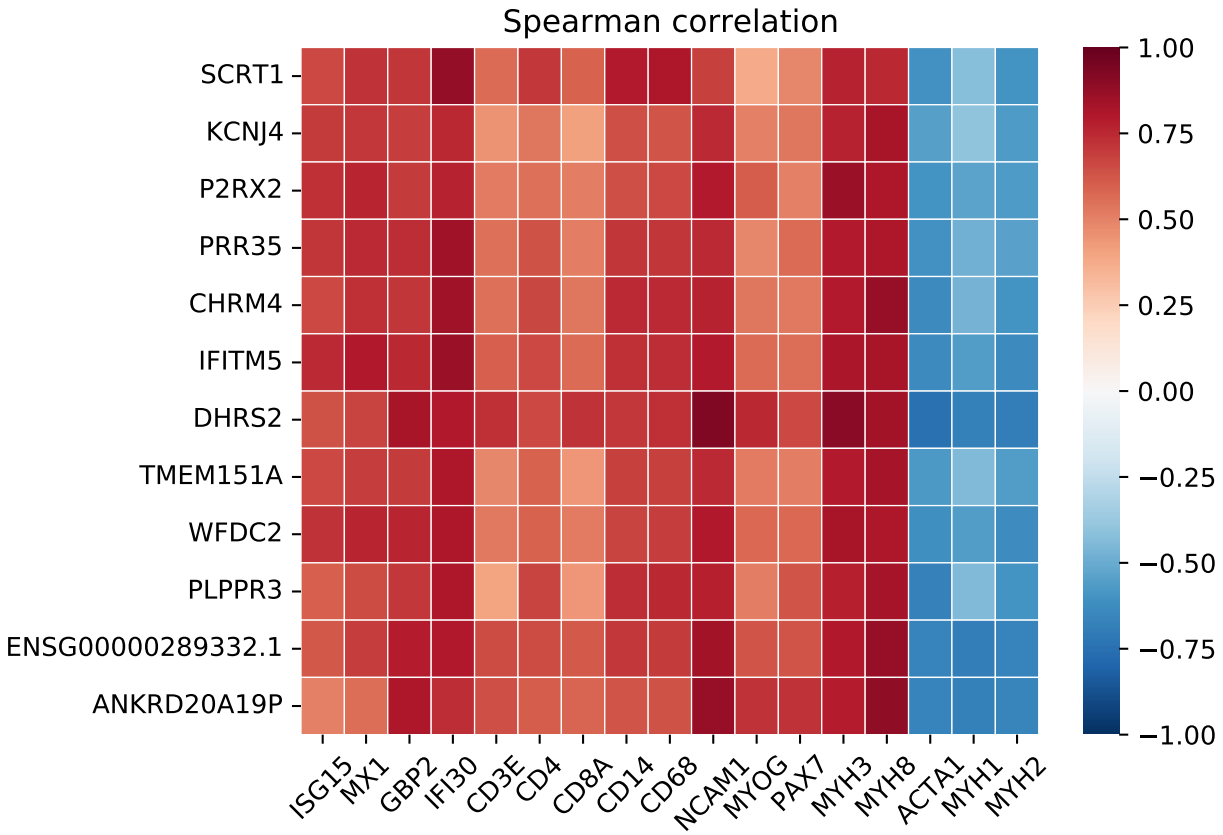


Figura 7. Ejemplo de *heatmap* de correlación

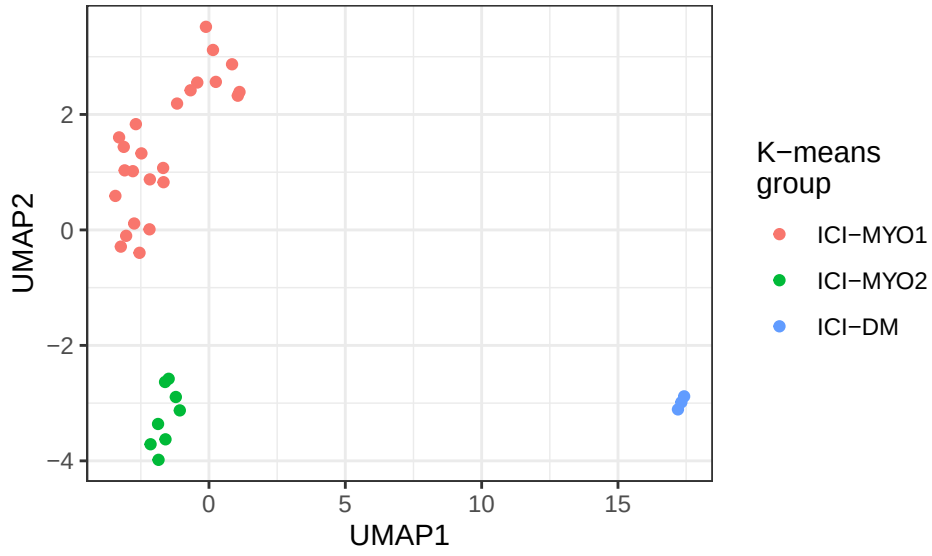


Figura 8. Ejemplo de aplicación de la *uniform manifold approximation for dimension reduction* (UMAP) para detectar la estructura de un grupo de muestras

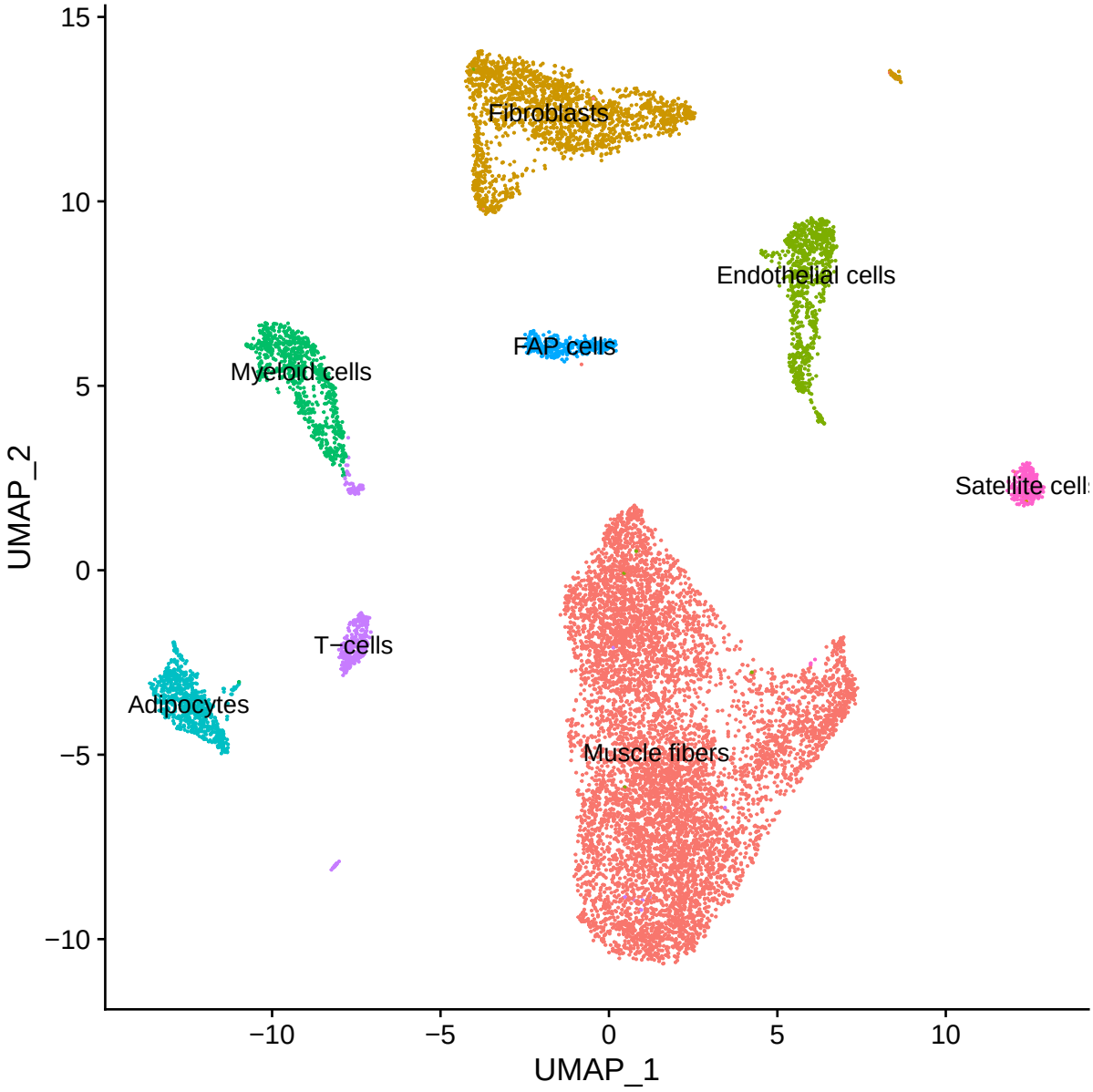


Figura 9. Ejemplo de aplicación de la *uniform manifold approximation for dimension reduction* (UMAP) para identificar los tipos celulares en una biopsia muscular

que si una vía determinada está afectada, sus genes tenderán a estar entre los más sobre o los más infraexpresados (Figura 10). El problema común fundamental de las técnicas de análisis de *pathways* es que están basados en una información incompleta y difícil de definir, los *pathways*. Por ejemplo, hay pathways que tienen un nivel de solapamiento muy elevado, como el IFN tipo I y el IFN tipo II pero que sin embargo están asociados a tipos de enfermedad totalmente diferentes. Asimismo, algunos de estos pathways se estudiaron en células o tejidos diferentes a los de interés, y esto no se tiene en cuenta en el análisis.

Como se puede deducir de lo explicado anteriormente, existen una serie de pasos que se pueden realizar en paralelo para cada muestra individual. Todos estos análisis individuales de cada muestra convergen en la generación de la tabla de cuentas. A partir de aquí hay una serie de dependencias entre análisis, pero el trabajo se puede paralelizar hasta cierto punto. Para realizar todos estos pasos de manera eficiente se requiere un sistema de gestión de dependencias. En los últimos años se han propuesto varias alternativas,(22) pero la solución más ampliamente disponible en todos los sistemas es la utilización de *GNU make*. GNU make permite generar listas de tareas con dependencias y productos necesarios que sólo se ejecutarán si las dependencias han sido modificadas.

El segundo problema a abordar en este tipo de análisis es la reproducibilidad de los mismos en diferentes entornos operativos. Este problema se puede dividir en dos, uno es la variabilidad de librerías usadas con los diferentes lenguajes de programación utilizados, y la segunda es la heterogeneidad de sistema operativo, lo que puede hacer que una misma librería no funcione de la misma manera. Para esto también hay múltiples alternativas, como el uso de *environments* con instalaciones locales de librerías o el uso de máquinas virtuales. No obstante, el uso de *environments* no soluciona el problema de la heterogeneidad de sistemas operativos y el uso de maquinas virtuales hace, generalmente, un uso poco eficiente de los recursos de computación. Una solución que puede abordar los dos problemas de manera óptima es el uso de *containers*, que son piezas de software que comparten el *kernel* del

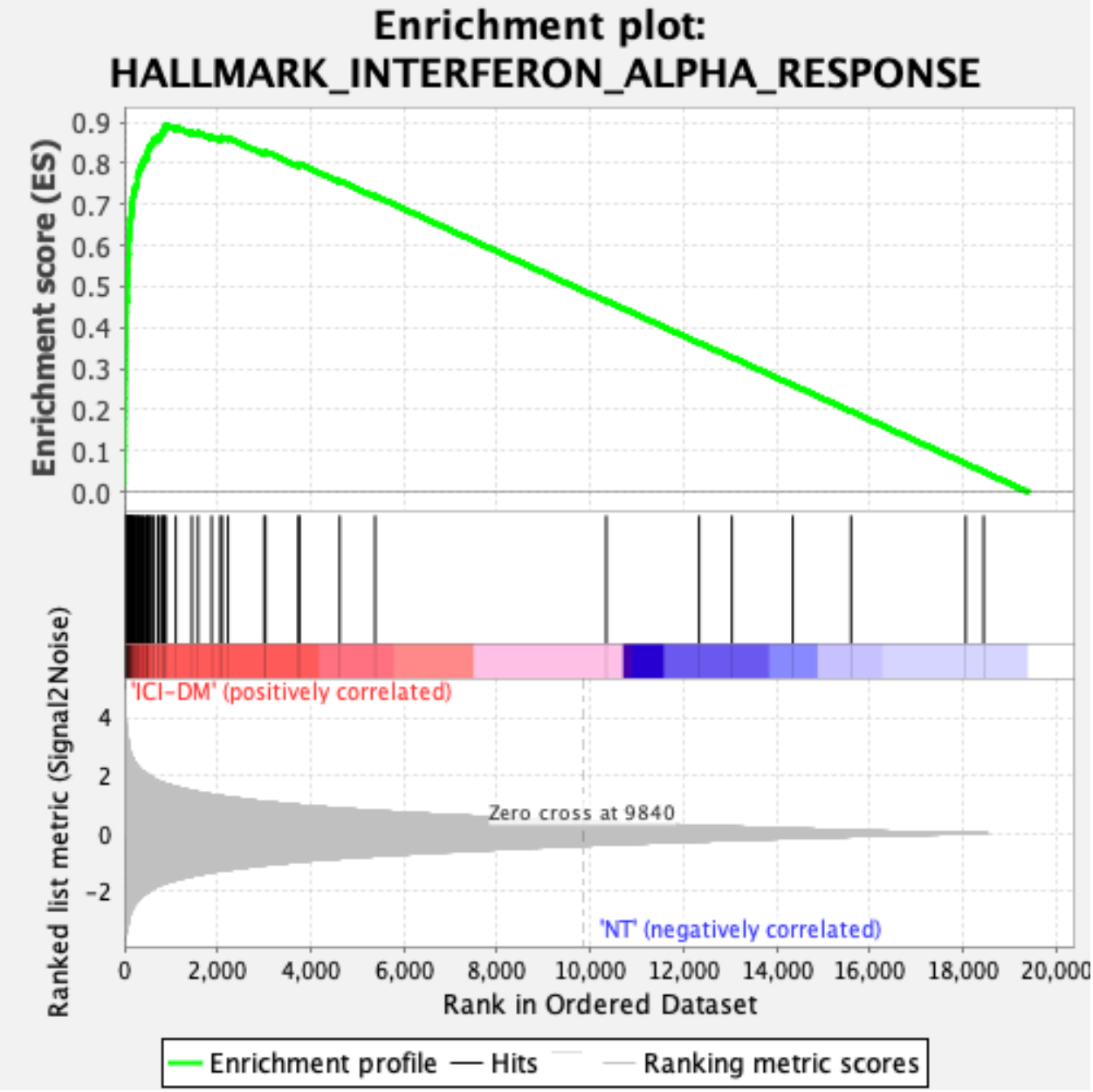


Figura 10. Ejemplo de análisis de *Gene Set Enrichment Analysis* (GSEA)

sistema operativo en el que se ejecutan, ofreciendo un sistema operativo virtual que aísla de manera eficiente el entorno de programación. Existen múltiples sistemas para el uso de container, incluido el popular *docker*, *podman*, o *singularity.*(23)

Además de realizar análisis dirigidos para responder preguntas concretas, es necesario precisar realizar múltiples pequeños análisis exploratorios. Poder disponer de un sistema de gestión y visualización de los resultados, facilita la generación de ideas y el transformar estás ideas en conocimiento útil. Un sistema de visualización idealmente necesita ser: 1) sencillo de mantener y actualizar para optimizar el tiempo del que lo mantenga, 2) fácil de ejecutar para que lo puedan usar colegas con pocos conocimientos técnicos y 3) disponer de una batería de análisis lo suficientemente amplia y flexible.

Objetivos

- El primer objetivo de este trabajo es diseñar un sistema para automatizar pipelines de secuenciación de RNA en sistemas distribuidos de manera que se puedan utilizar de manera dinámica y modificar con facilidad a medida que desarrollen nuevas herramientas o se actualicen las que se usan en este momento.
- El segundo objetivo es desarrollar un software de visualización de este tipo de datos para poder realizar análisis exploratorios de manera eficiente.

Implementación

En esta sección se detallará la implementación de las dos secciones de este trabajo fin de carrera: a) el diseño de un pipeline de secuenciación de RNA en sistemas distribuidos (Apéndices 1.1-3) y b) el desarrollo de un software de visualización para análisis exploratorio de datos (Apéndice 1.4).

En cuanto a la implementación del pipeline de secuenciación se dividió en dos pasos, la demultiplexación (Apéndice 1.1) y la generación de cuentas (Apéndice 1.2). El motivo de esta estructura es que tras la demultiplexación hay que realizar un control e calidad del archivo resultante de forma manual y, a menudo, resecuenciar la muestra si ha habido artefactos técnicos o un número insuficiente de lecturas totales.

Demultiplexación

Se generó un archivo Makefile y un script bash (demux.sh) para demultiplexar datos de secuenciación de RNA crudos utilizando el software bcl2fastq. El archivo Makefile define varios objetivos y dependencias, mientras que el script de bash realiza el demultiplexado real. bcl2fastq es una imagen de Singularity, lo que permite su reemplazo con una nueva versión o con otra herramienta de manera sencilla. Este proyecto permite crear nuevos *batches* de secuenciación, borrarlos, y obtener las lecturas de secuenciación resultantes de manera eficiente, permitiendo la ejecución en paralelo de diferentes *batches* de secuenciación y el uso de varios hilos de demultiplexación por cada *batch* (Apéndice 1.1).

Alineamiento y cuantificación

Para el alineamiento y cuantificación (Apéndice 1.2) se optó por un proyecto make que organiza el arbol de dependencias para que se ejecuten los diferentes fragmentos de código de manera eficiente.

El proyecto requiere un archivo de metadatos en que se indica el nombre de cada muestra y la localización de los archivos *.fastq* resultantes de la demultiplexación y un proyecto git en que se almacena toda la base de código de los diferentes proyectos y al que se accede para obtener cualquier archivo de código que no esté presente en el proyecto actual.

En primer lugar, se descarga la base de código del repositorio de GitHub y se indica cómo obtener cualquier archivo *.sh*, *.py* o *.R* del proyecto. Posteriormente, se genera una lista de muestras a secuenciar y el listado de tablas de expresión para cada muestra (archivo *quant.sf* de cada muestra).

Dado que nuestros archivos de secuenciación se han generado utilizando diferentes parámetros de secuenciación, además de limpiar los archivos fastq es necesario adaptarlos a los requerimientos menos exigentes de cada conjunto de muestras, para evitar artefactos técnicos. Desde un punto de vista práctico esto implica decidir si el conjunto de muestras debe tener una o dos lecturas y cuál debe ser su longitud. Para realizar esto se generaron dos scripts, *get_len_fastq.py* y *keep_r2.py*, el primero para calcular la longitud de la secuencia a partir de las primeras mil lecturas de cada fastq y el segundo para detecta si todas las muestras tenían dos lecturas *paired_end* o sólo una *single_end*. El resultado de ejecutar estos scripts en el producto de la demultiplexación de todas las muestras se almacena en un archivo *seq_vars_file* que se cargará en el entorno cuando sea necesario.

El primer paso del *pipeline* de análisis es obtener los archivos demultiplexados del subgrupo de muestras a analizar (en este caso todas las muestras de secuenciación de RNA en bloque), para esto se generó un script, *get_fastq.py* que utiliza los parámetros almacenados

en *seq_vars_file* para realizar accesos directos al archivo, o, si el archivo necesario proviene de varios *runs* de demultiplexación, generar el *.fastq* combinando los diferentes *runs*.

Tras haber obtenido los archivos *.fastq* necesarios, estos se limpian y se realiza un primer control de calidad usando *fastp*, que es una herramienta que, además de combinar limpieza de lecturas y control de calidad, realiza ambas tareas de manera eficiente.

Tras esto se procede a la alineación y generación de tabla de cuantificación de transcritos. Pero para realizar este paso es primero necesario tener disponible el genoma de referencia de la especie (en este caso el genoma de referencia humano), almacenado en *genome.fa.gz*, la anotación de los transcritos, almacenado en *transcripts.fa.gz*, y el archivo *.gtf* donde se definen las coordenadas genómicas de cada gen y tránscrito. Además, también es preciso obtener archivos de anotación para poder traducir entre código de Ensembl de dada gen y su símbolo oficial. Tras descargar esta anotación se generan archivos de anotación con el *script get_annotation.R*.

Para la alineación y cuantificación se decidió usar la herramienta Salmon, que es un pseudoalineador que tiene como ventaja principal la velocidad de procesamiento y obtención de resultados. Esto es crítico para nuestros proyectos en que es frecuente tener más de medio millar de muestras a analizar. Para utilizar esta herramienta es necesario primero generar sus índices, lo que se realiza con *salmon_index.sh*.

Salmon, a través del script *salmon.sh*, genera un archivo de cuentas *quant.sf* de todos los tránscritos de cada muestra. El producto de este paso de cada muestra se fusionará en una única tabla de expresión a nivel de genes, *gene_counts.csv* utilizando la herramienta *tximport* por medio del *script tximport.R*.

Ahora que ya tenemos todos los productos necesarios es preciso realizar un control de calidad. Dado el volumen de muestras con el que trabajamos, explorar cada una manualmente no es práctico y es necesario poder agregar los estadísticos de calidad, lo que se puede hacer con la herramienta MultiQC.

En este momento se examinarán manualmente los resultados del control de calidad de cada muestra y se decidirá si hay muestras que necesitan resecuenciarse para obtener más lecturas o de las que tiene que volver a repetirse la preparación de la librería por problemas técnicos.

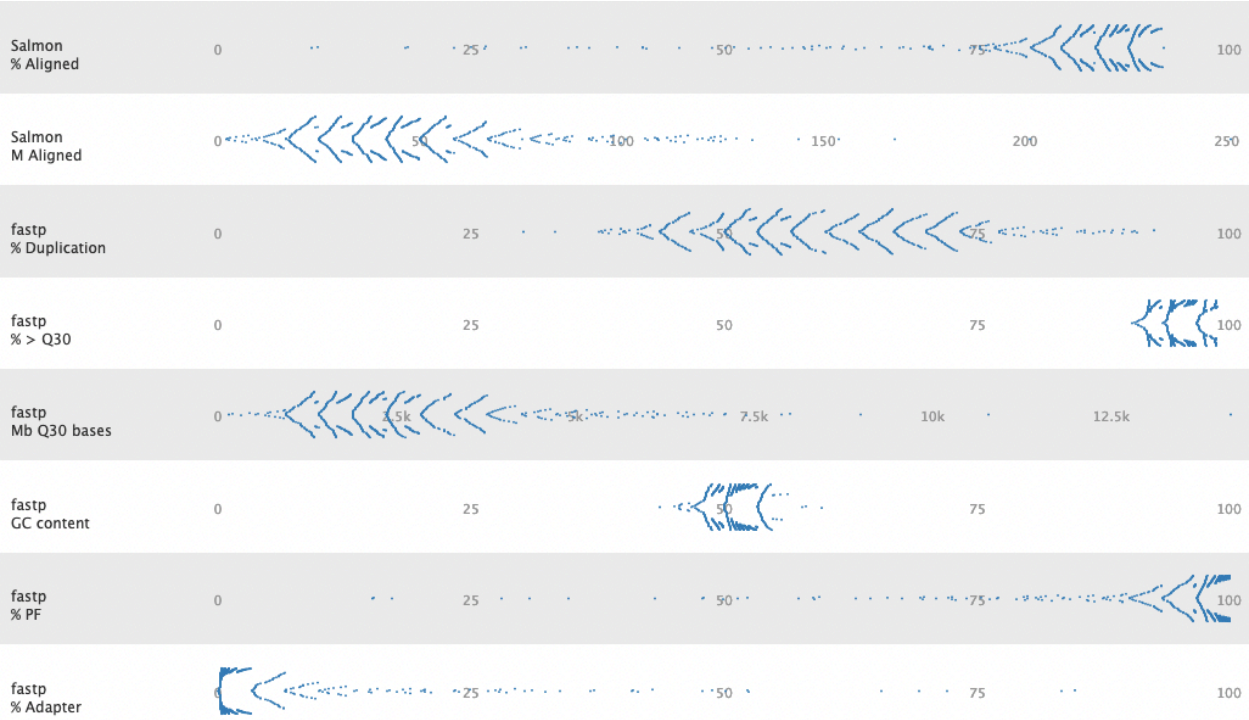


Figura 11. Ejemplo de sección del control de calidad con MultiQC

Una vez obtenida la matriz de cuentas de todas las muestras para cada gen, se procesa este archivo de cuentas para realizar normalizarlas y obtener los TMMs utilizando el paquete EdgeR. Además, en vez de tener los genes en las filas y las muestras en las columnas, para realizar determinados análisis gráficos es más sencillo tener los genes en las columnas y las muestras en las filas. Para lograr estos dos objetivos se diseñaron dos scripts, *counts_to_tmm.R* que normaliza la matriz de cuentas a TMMs, y *setup_tmm.R* que transpone las cuentas para que los genes estén en las columnas.

Con esto se finaliza el preprocesamiento de las muestras, que, para este tipo de análisis es la parte para la que es más necesario procesar los datos en paralelo a través de un clúster de computación.

Expresión diferencial, análisis de pathways y análisis gráfico

Una vez obtenida la matriz de cuentas, habitualmente el primer paso del análisis es realizar el estudio de expresión diferencial, o, en otras palabras, comparar la expresión de todos los genes en los diferentes grupos de interés. Para esto, hay una serie de análisis que suelen ser necesarios y pueden automatizarse hasta cierto punto (Apéndice 1.3).

En primer lugar, cuando hay una variable categórica que define los grupos de interés, es habitualmente necesario compara todos los grupos entre ellos, y cada uno con el resto de las muestras. Además, es habitualmente conveniente que el grupo de referencia sea el segundo del análisis para que así el *fold-change*, o número de veces que un gen esté más expresado en un grupo que en el otro, sea fácilmente interpretable. Además es frecuentemente necesario realizar una serie de operaciones de preprocesamiento en este tipo de análisis, como por ejemplo restringir las comparaciones a aquellas que contengan un grupo concreto, o filtrar las muestras. Finalmente, es habitual disponer de variables confusoras que quieren incluirse como variables de ajuste en el análisis. Para lograr todos estos objetivos se diseñó un script, *limma.R*, que implementa todas estas funcionalidades de manera sencilla y se puede utilizar tanto como un *script* de *R* o como un programa de línea de comandos. El producto final de ejecutar este *script* es una tabla de expresión diferencial ordenada por nivel de significación, como la que se muestra a continuación:

	logFC	AveExpr	t	P.Value	adj.P.Val	B
SCRT1	6.3	-5.1	18.8	5.86E-44	1.94E-39	87.8
P2RX2	5.4	-5.0	17.0	4.94E-39	8.16E-35	76.6
KCNJ4	5.5	-5.0	16.6	5.68E-38	6.26E-34	74.4
PRR35	5.4	-5.3	15.7	1.72E-35	1.42E-31	68.7
IFITM5	5.3	-5.6	15.3	2.83E-34	1.87E-30	65.9
DHRS2	5.1	-3.9	14.6	3.34E-32	1.84E-28	61.8
CHRM4	4.9	-3.6	14.5	5.79E-32	2.73E-28	61.3
PLPPR3	4.3	-5.2	13.6	2.75E-29	1.14E-25	54.8
ENSG00000289332.1	4.3	-5.3	13.2	2.92E-28	1.01E-24	52.5
WFDC2	4.2	-2.6	13.2	3.06E-28	1.01E-24	53.1

En esta tabla se obtiene el *fold-change* entre los dos grupos en escala logarítmica (*logFC*), la expresión media de dicho gen en los grupos a comparar (*AveExpr*), el estadístico T absoluto (t), el p-value, el p-value ajustado por Benjamini and Hochberg y las *odds* de expresión diferencial en escala logarítmica (B).

Tras realizar el estudio de expresión diferencial es a menudo necesario estudiar si los genes con expresión diferencial tienen algún tipo de función biológica común. Para esto se diseñó el script *cluster_profiler.R* que utiliza la librería clusterProfiler para calcular el análisis de *pathways* utilizando GSEA y la base de datos Reactome (Figura 12, Apéndice 1.3).

Finalmente, hay una serie de análisis gráficos que son a menudo necesarios (Apéndice 1.3). Para implementarlos se diseñaron dos librerías de funciones, una en R, *rnaseq_tools.R* y otra en python, *rnaseq_tools.py*.

La librería de R *rnaseq_tools.R* contiene funciones para generar gráficos individuales de cada gen (*individual_plot()*), heatmaps con los valores de z-score (*heatmap_z_score()*), además de contener funciones para descargar la lista de genes de comité de nomenclatura de genes

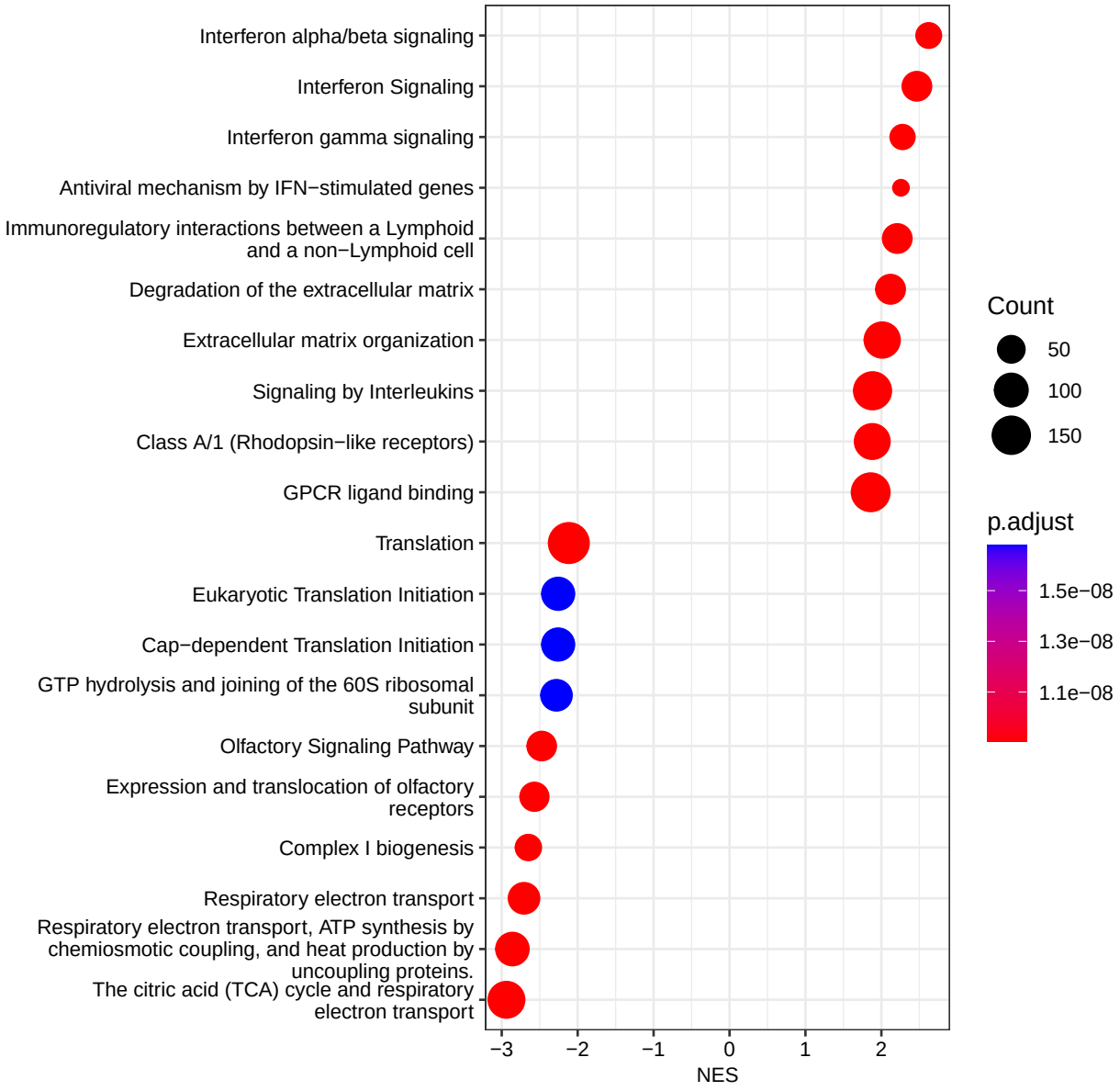


Figura 11. Ejemplo de representación gráfica de análisis de pathways usando GSEA, clusterProfiler y la base de datos Reactome

Hugo y aislar la lista de genes de una familia determinada de la base de datos Hugo.

Por su parte la librería de python *rnaseq_tools.py* continene 15 funciones diferentes, fundamentalmente para realizar análisis gráfico de los datos resultantes:

- whiskerplot_gene: Genera un gráfico para cada gen separado por grupo data una variable categórica *grouping_var*. Permite, dependiendo de la variable *style*, o bien generar un diagrama de cajas y un *swarmplot*, hacer un diagrama con los intervalos de confianza de Tukey, o hacer un diagrama de líneas.
- whiskerplot_group: Genera un gráfico para múltiples genes separados en las diferentes categorías de una variable *group*.
- heatmap_genes_by_group: Realiza un *heatmap* de *fold-change* de las categorías de una variable de agrupación relativas a su valor normal "NT".
- heatmap_genes_longitudinal: Realiza un gráfico en que el fold-change se muestra longitudinalmente para los diferentes valores de la variable *longitudinal_var*.
- heatmap_genes_correlation: Función que calcula el coeficiente de correlación de Spearman entre dos grupos de variables *horizontal_gene_set* y *vertical_gene_set*.
- bivariate_scatter: Función que realiza un gráfico de puntos bivariado entre dos variables *gene1* y *gene2*.
- compact_flatten_hugo_list: Función que filtra y aplana la lista de genes de Hugo en base a una familia determinada.
- filter_limma: Devuelve el resultado de la expresión diferencial para una lista de genes.
- top_genes_limma: Devuelve una tabla con los genes más significativamente expresados de una tabla de expresión diferencial.
- list_genes_limma: Genera una tabla de expresión diferencial formateada para publicación dada una lista de genes.

- top_genes_per_group_graph: Genera un gráfico con la expresión de los 10 genes más significativos de una comparación determinada.
- multi_gene_graph: Genera un gráfico combinado de varios genes formateando automáticamente la apariencia para poder utilizarse para publicación.
- correlation_gene: Función que crea una tabla de correlación de los genes de interés con el resto de los genes del estudio.
- gen_groups: Función para generar los grupos más habituales en nuestros estudios en base a una variable con una cadena de texto.
- setup_gene_tmm: Transpone la tabla de genes para poder realizarse en análisis gráfico utilizando esta librería.

Como se puede observar, con esta librería de funciones se pueden generar los análisis gráficos indicados en la introducción.

Visualizador

Una necesidad fundamental al realizar análisis exploratorio de grandes bases de datos es poder visualizar grandes volúmenes de datos de manera sencilla. Esto permite comprobar hipótesis y priorizar objetivos de investigación de manera eficiente.

Un visualizador de datos eficiente en nuestra área de investigación debe poder realizar una serie de tareas. Entre ellas, generar datos de diferentes proyectos de manera sencilla, generar visualizaciones de genes individuales categorizados por una o dos variables, generar gráficos de puntos dadas dos variables cuantitativas de interés, filtrar muestras y permitir visualizar los valores numéricos de cada gen para todas las muestras, identificar los genes más correlacionados con el gen de interés, realizar estudios exploratorios de la calidad de los genes, y obtener información de bases de datos externas del gen en cuestión.

Para realizar todos estos objetivos se decidió usar la librería de R Shiny (Apéndice 1.4). Esta librería divide la funcionalidad en el *backend* (server.R) y el *frontend* (ui.R) de la aplicación, permitiendo realizar sistemas modulares que se pueden desarrollar y mantener de manera sencilla.

El frontend de la aplicación permite seleccionar el proyecto en el que estamos trabajando. Para añadir proyectos al visualizador sólo es necesario realizar un acceso directo a la tabla de TMMs transpuesta en la carpeta de datos del proyecto e indicar en el archivo sort_order.txt el orden de las diferentes categorías de interés, si lo hubiese. Además, permite seleccionar un grupo de interés para visualizar en el eje de abscisas, un gen de interés para mostrar en escala logarítmica en el eje de ordenadas, una variable para mostrar como color en los puntos del valor de las observaciones, y un segundo gen para estudios de correlación. Finalmente, permite refrescar la base de datos cuando se añade un nuevo proyecto.

En cuanto a los análisis que se han implementado en la versión actual, se puede realizar análisis gráficos estratificados por una variable categórica y coloreados por una segunda variable categórica (ventana *Plot*, Figuras 12 y 13), visualizar los datos crudos (*Raw*, Figura 14), visualizar la relación entre dos variables (*Two-gene comparisons*, Figura 15), estudiar la correlación del gen actual con todos los otros genes del estudio (*Gene correlation*, Figura 16), realizar control de calidad por PCA (*Quality control*, Figura 17) y obtener la información de cada gen de *genecards.org* (*Gene info*, Figura 18).

Todos los datos generados en este visualizador se pueden exportar de manera sencilla para enviar a colaboradores o usar en diferentes proyectos.

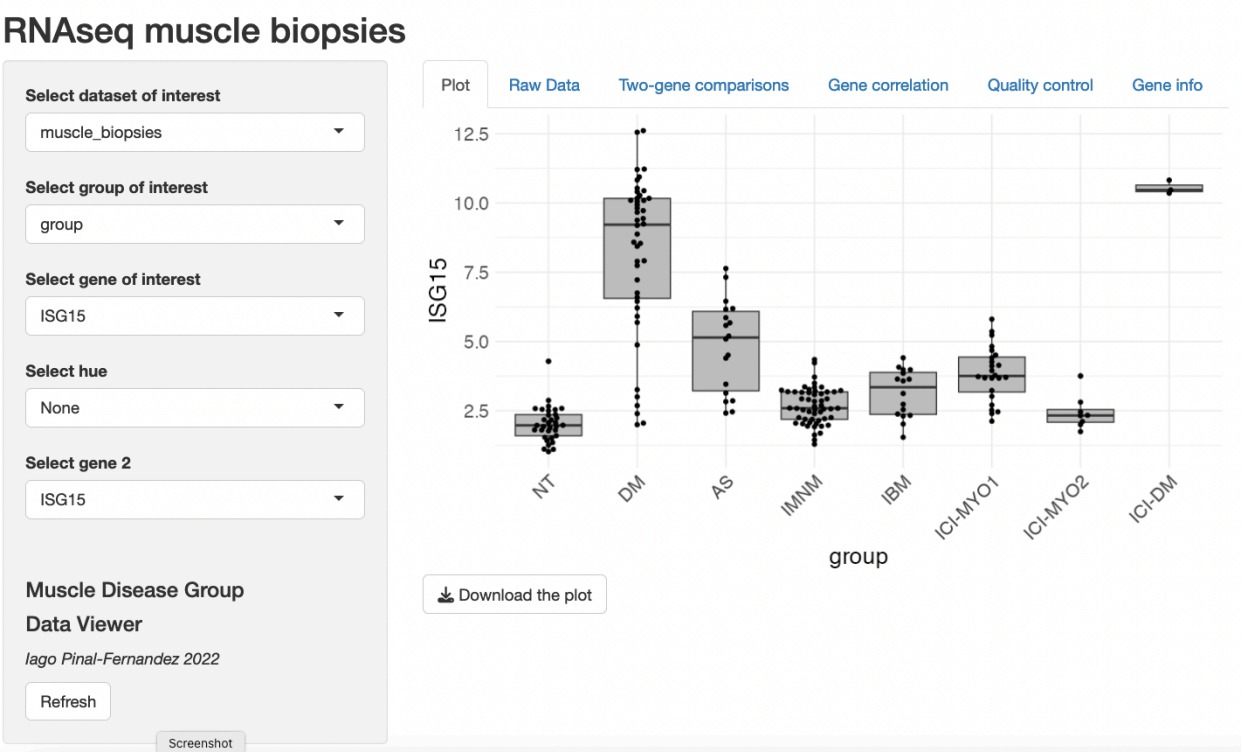


Figura 12. Análisis gráfico estratificado por grupo clínico

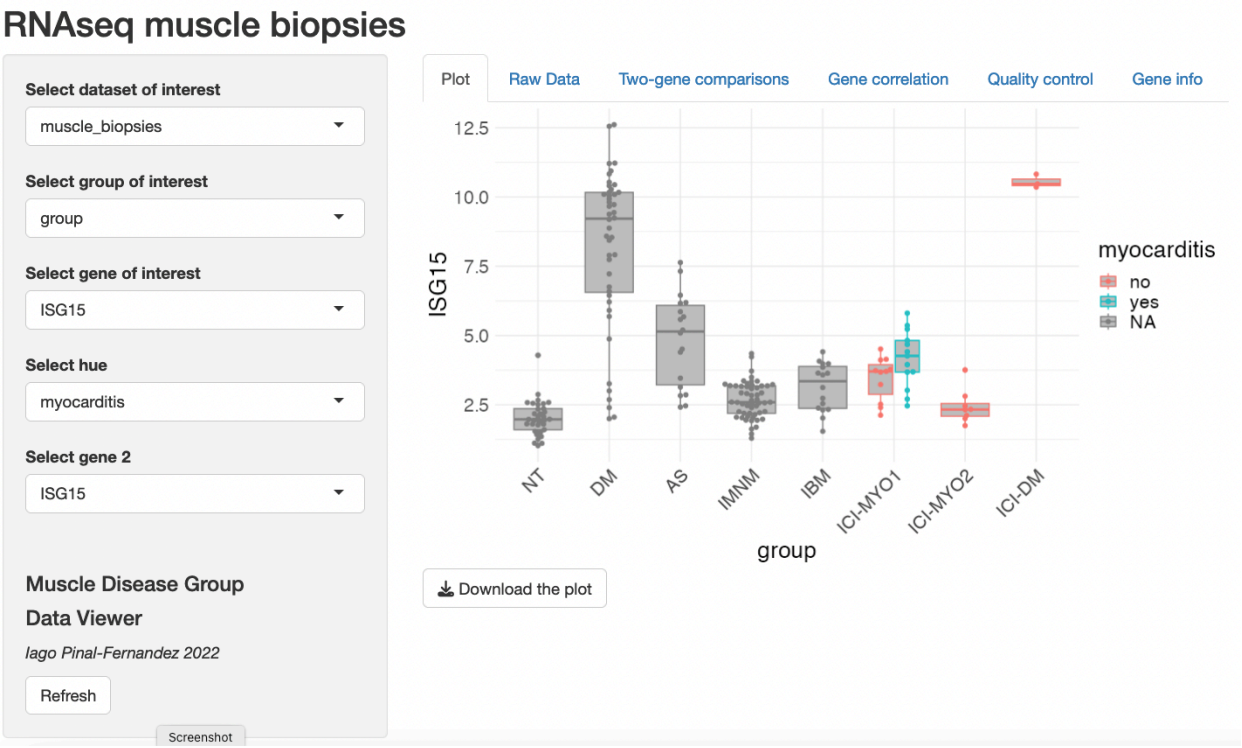


Figura 13. Análisis gráfico estratificado por grupo clínico y coloreado en base a la presencia de miocarditis

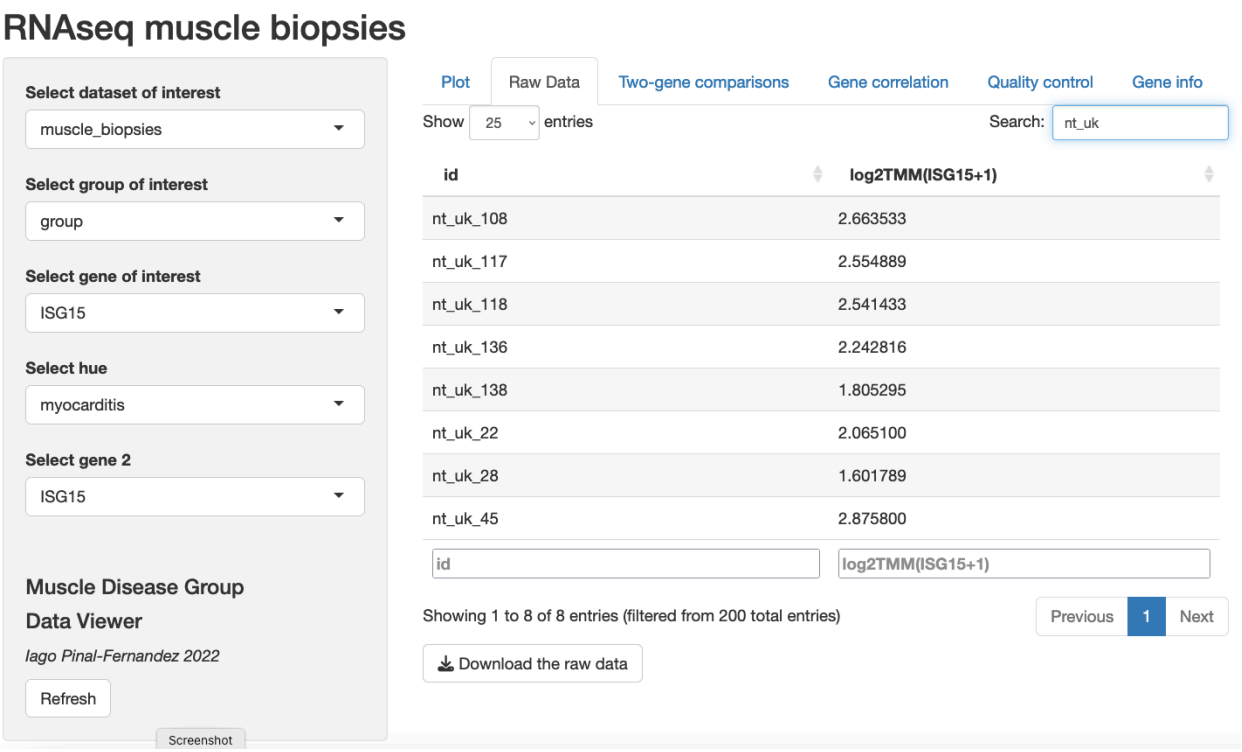


Figura 14. Valores crudos de uno de los genes filtrados por nombre de muestra

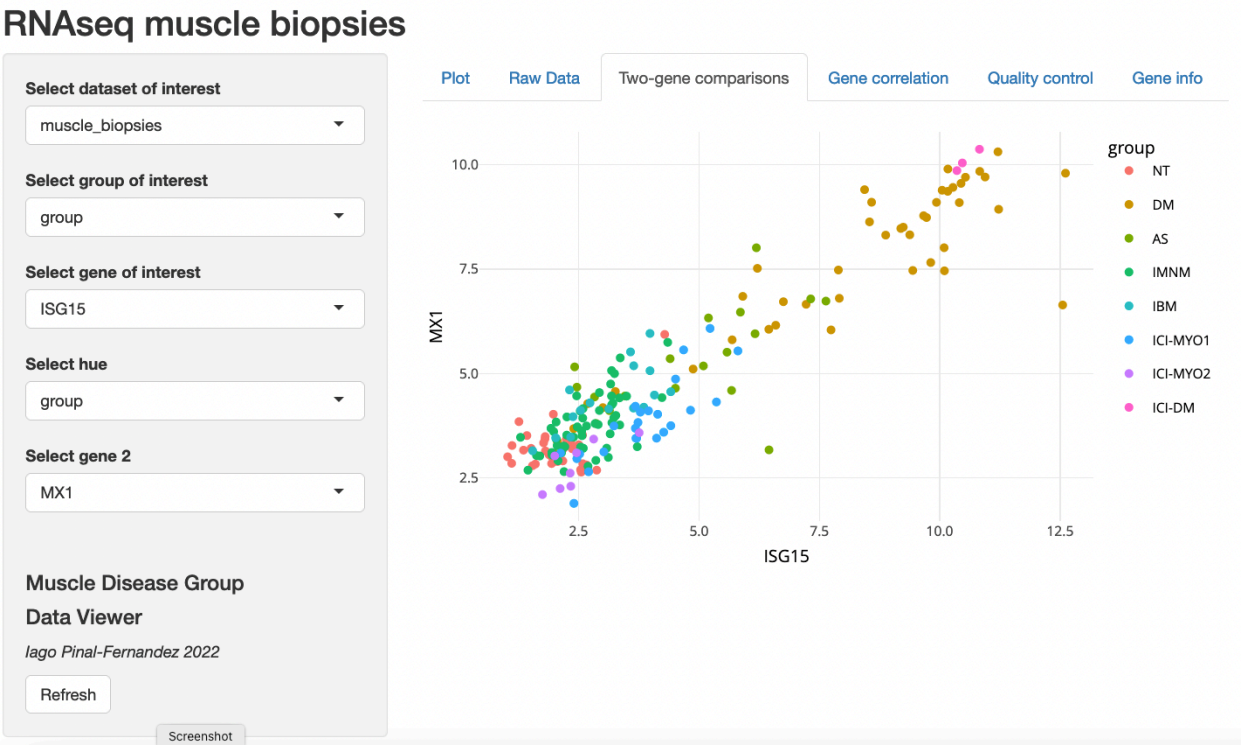


Figura 15. Gráfico bivariado de dos genes coloreado por grupo clínico

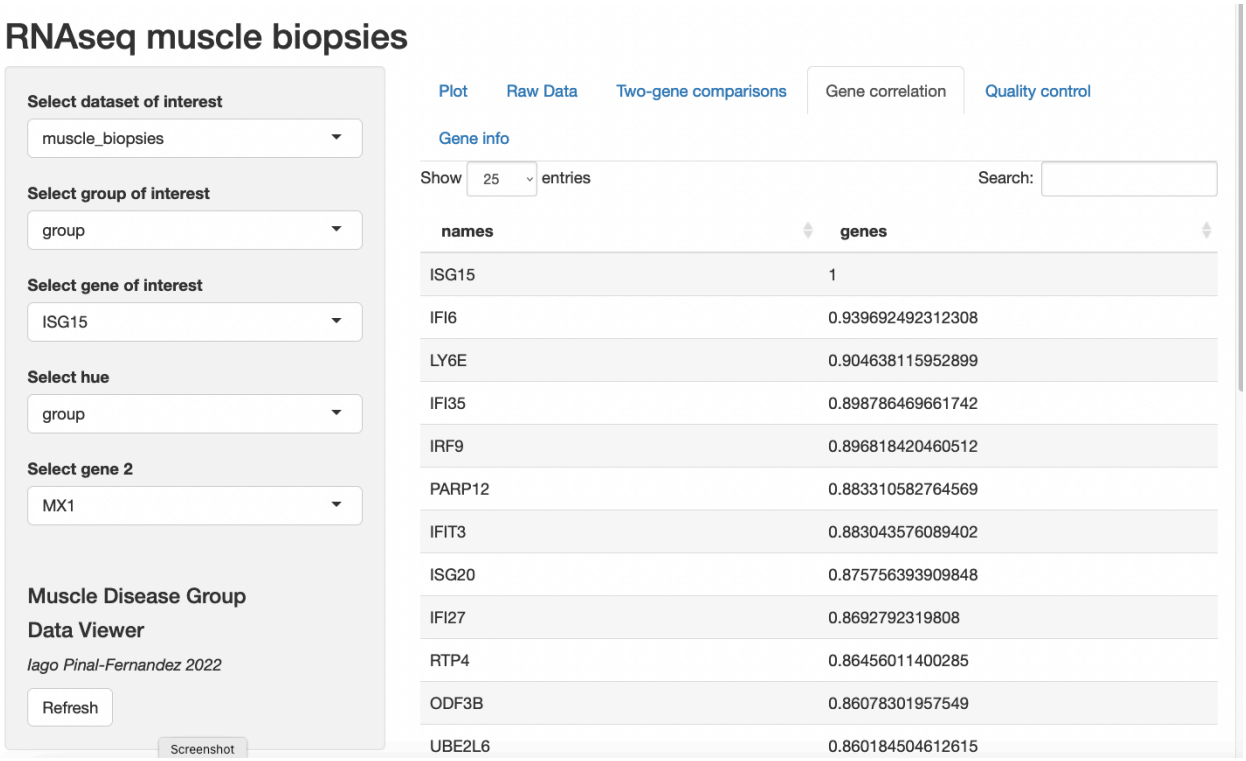


Figura 16. Genes correlacionados con nuestro gen de interés

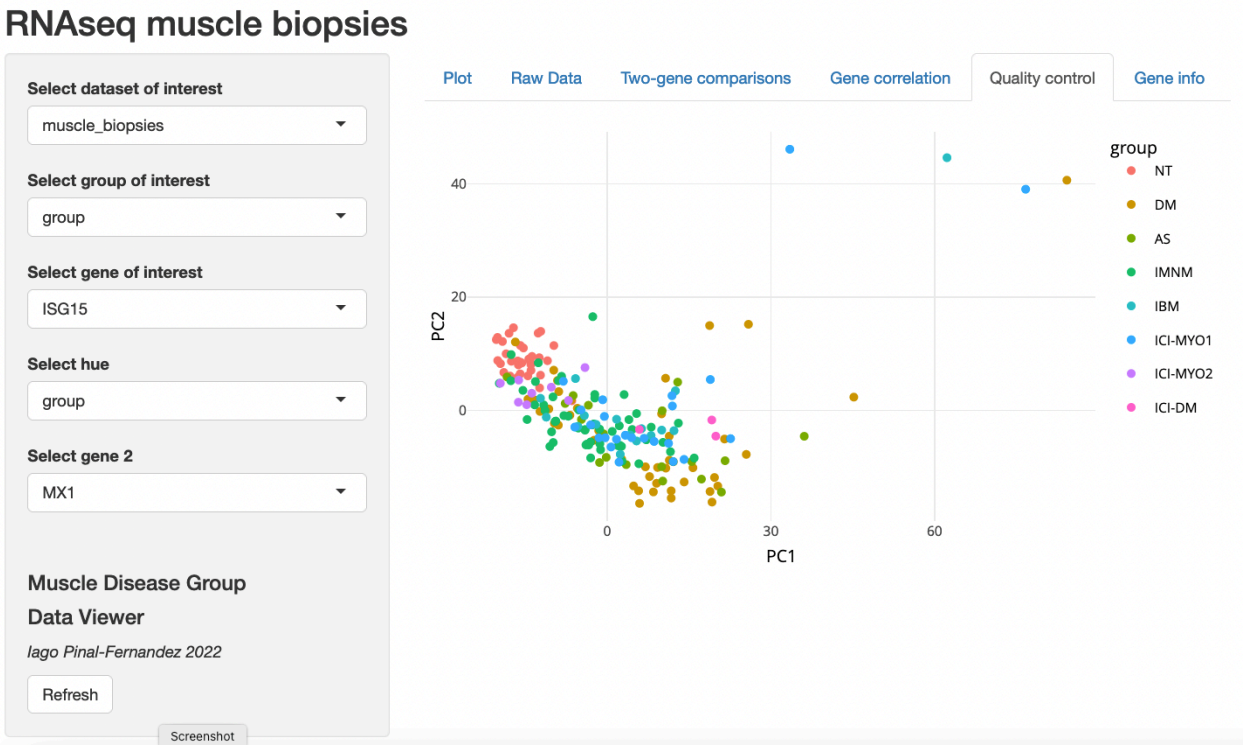


Figura 17. Reducción de dimensionalidad usando PCA para control de calidad

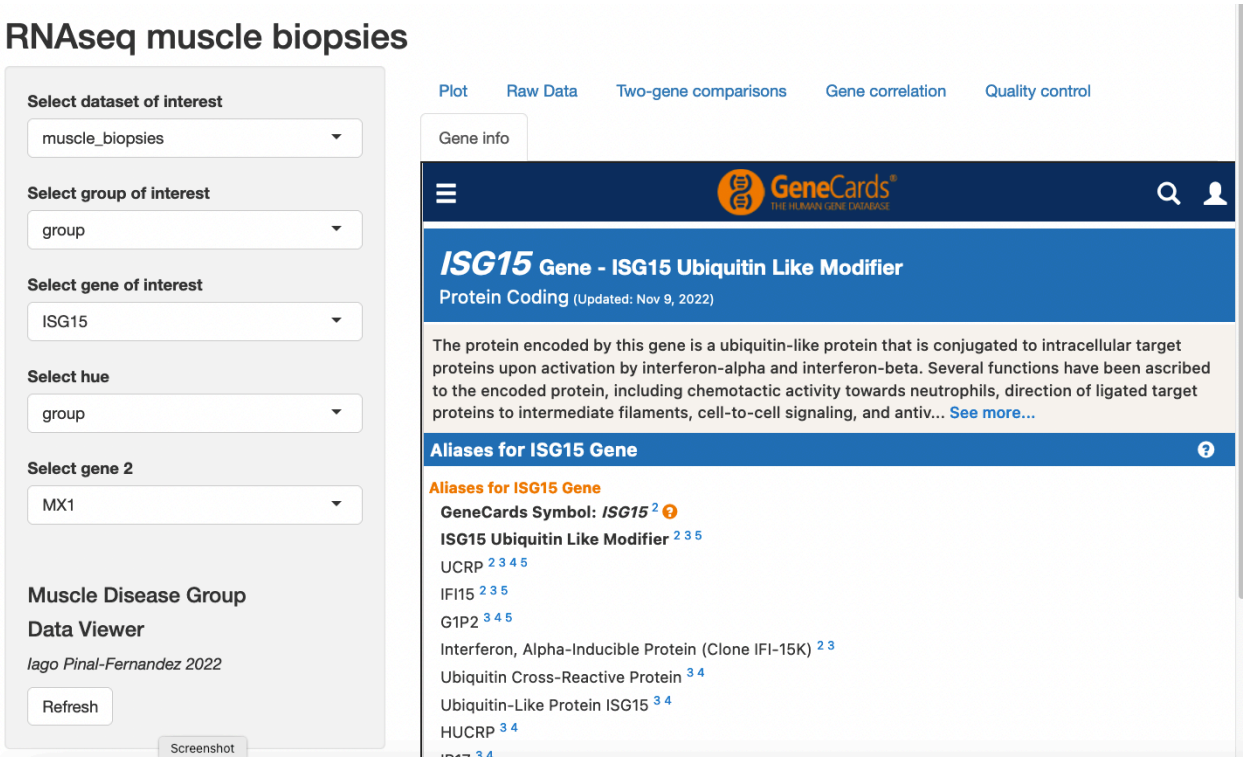


Figura 18. Visualización de la informaciónd de uno de los genes

Discusión

En este proyecto, y de acuerdo con los objetivos planteados inicialmente, primero se diseñó un sistema escalable para automatizar pipelines de secuenciación de RNA en sistemas distribuidos (Apéndice 1.1-3) y en segundo lugar se desarrolló un software de visualización de este tipo de datos para poder realizar análisis exploratorios de manera eficiente (Apéndice 1.4).

Automatizar y estandarizar el procesamiento de datos de secuenciación de RNA es esencial para garantizar la calidad de los resultados del análisis. Además, esto permite una mayor eficiencia en el análisis y la interpretación de los datos, ya que se pueden aplicar automáticamente métodos estandarizados para la limpieza y el preprocesamiento de los datos.

Una de las principales ventajas de un sistema escalable es que permite el procesamiento de un gran volumen de datos, lo que es especialmente útil en estudios con elevado número de muestras, como los que estamos realizando. Además, al utilizar un sistema distribuido, se pueden aprovechar el procesamiento en paralelo para acelerar el procesamiento, lo que permite obtener resultados más rápidamente, una característica clave cuando se están realizando varios proyectos en paralelo.

Un software de visualización eficiente es también crucial para el análisis exploratorio de los datos. Los datos de secuenciación de RNA son complejos y pueden tener un gran número de dimensiones, lo que hace difícil la interpretación de los resultados. Un buen software de visualización puede ayudar a representar de manera visual y comprensible los resultados del

análisis, lo que permite a los investigadores identificar patrones y tendencias en los datos en tiempo real. Esto permite identificar de manera precoz que estudios confirmatorios pueden aportar más valor añadido y reduce tiempo de realizar estudios preliminares.

Gracias a las herramientas bioiformáticas realizadas durante este proyecto se pudieron generar de una manera eficiente tres publicaciones que están actualmente en proceso de revisión por pares en diversas revistas internacionales. El factor común de estas tres publicaciones es que utilizan como fuente generadora de hipótesis primaria los datos de secuenciación de RNA en bloque, utilizando como técnicas confirmatorias diversos estudios más especializados, como secuenciación de RNA de célula única o de núcleo único.

En concreto, el primer estudio (Apéndice 2.1) definió el perfil transcripcional del músculo de pacientes con cáncer que desarrollan miopatía tras recibir tratamiento quimioterápico con un tipo de fármacos denominados *checkpoint inhibitors* que activan el sistema inmune y, característicamente, geneneran fenomenos autoinmunes como efecto adverso. En este estudio identificamos tres grupos diferentes de pacientes con este tipo de miopatía.

- El primer grupo de pacientes (ICI-DM) tenía afectación cutánea similar a la dermatomiositis y anticuerpos anti-TIF1g. Al igual que pacientes con dermatomiositis clásica, tenían niveles muy elevados de interferón de tipo 1.
- El segundo grupo de pacientes (ICI-MYO1) tenía biopsias altamente inflamatorias y incluyó a todos los pacientes que desarrollaron miocarditis (inflamación del músculo cardiaco).
- El tercer grupo (ICI-MYO2) incluyó a pacientes con necrosis de fibras musculares y niveles bajos de inflamación muscular.

Además se identificó que todas las biopsias tenían elevación de la vía de la IL6 y que tanto ICI-DM como en ICI-MYO1 tenían elevación de la vía del interferón tipo 2. Tanto la vía del interferón tipo 1, como la vía del interferón tipo 2 y la vía de la interleukina 6 se

pueden modular usando diferentes tratamientos y estos resultados pueden tener implicaciones directas en el manejo de estos pacientes.

En el segundo estudio (Apéndice 2.2) se estudió sistemáticamente la expresión de diferentes genes del complemento en pacientes con miopatía inflamatoria, determinando que estos genes están sobreexpresados de manera generalizada en miostis, que diferentes tipos de células expresan diferentes subconjuntos de genes y que el interferón gamma es capaz de inducir esta expresión.

Finalmente, el último estudio (Apéndice 2.3) identificó que los pacientes con dermatomiosits y anticuerpos anti-Mi2 tienen un perfil transcripcional característico con derepresión de más de 100 genes con funciones celulares heterogéneas. El nivel de expresión de estos genes está mutuamente correlacionado, asociado a los niveles de autoanticuerpos de los pacientes, y ligado a la actividad de la enfermedad. Esto es importante porque los anticuerpos anti-Mi2 reconocen una serie de proteinas que son miembros del complejo NuRD, que es un complejo represor de otros genes. Nuestros resultados sugieren que este tipo de miopatía puede estar causada por una alteración del complejo NuRD por los autoanticuerpos de los pacientes, generando una derepresión transcripcional que puede ser tóxica para las células musculares.

La realización de estos tres estudios en un periodo tan limitado de tiempo no habría sido posible sin el esfuerzo de estandarización y automatización realizados durante este proyecto fin de grado.

Pero este es un proyecto aún en desarrollo.

- En primer lugar, para la sección de demultiplexación se generó un contenedor para ejecutar el software especializado necesario pero el resto del proyecto aún requiere instalaciones locales que son difícilmente replicables en otros entornos. En un futuros esperamos poder generar contenedores para todas las herramientas que utilizamos, haciendo el sistema dependiente sólo de la herramienta de generación y uso de los contenedores.

- En segundo lugar, el sistema de visualización todavía se puede expandir para añadir otras funcionalidades, como la habilidad filtrar listas de muestras, generar gráficos de listas de genes, completar el control de calidad o implementar el estudio de pathways.

A pesar de estas limitaciones, en este proyecto se ha podido hacer un avance significativo en este área de la computación, demostrando de manera robusta la utilidad de las herramientas generadas para generar conocimiento científico de manera eficiente.

Referencias

1. Selva-O'Callaghan A, Pinal-Fernandez I, Trallero-Araguás E, Milisenda JC, Grau-Junyent JM, Mammen AL. Classification and management of adult inflammatory myopathies. The Lancet Neurology. 2018 Sep;17:816–28.

2. Casal-Dominguez M, Pinal-Fernandez I, Pak K, Huang W, Selva-O'Callaghan A, Albayda J, et al. Performance of the 2017 european alliance of associations for rheumatology/american college of rheumatology classification criteria for idiopathic inflammatory myopathies in patients with myositis-specific autoantibodies. Arthritis & rheumatology (Hoboken, NJ). 2022 Mar;74:508–17.

3. Pinal-Fernandez I, Casal-Dominguez M, Derfoul A, Pak K, Plotz P, Miller FW, et al. Identification of distinctive interferon gene signatures in different types of myositis. Neurology. 2019 Sep;93:e1193–204.

4. Greenberg SA, Pinkus JL, Pinkus GS, Burleson T, Sanoudou D, Tawil R, et al. Interferon-alpha/beta-mediated innate immune mechanisms in dermatomyositis. Annals of neurology. 2005 May;57:664–78.

5. Pinal-Fernandez I, Casal-Dominguez M, Derfoul A, Pak K, Miller FW, Milisenda JC, et al. Machine learning algorithms reveal unique gene expression profiles in muscle biopsies from patients with different types of myositis. Annals of the rheumatic diseases. 2020 Sep;79:1234–42.

6. Pinal-Fernandez I, Amici DR, Parks CA, Derfoul A, Casal-Dominguez M, Pak K, et al. Myositis autoantigen expression correlates with muscle regeneration but not autoantibody specificity. Arthritis & rheumatology (Hoboken, NJ). 2019 Aug;71:1371–6.

7. Ikenaga C, Date H, Kanagawa M, Mitsui J, Ishiura H, Yoshimura J, et al. Muscle transcriptomics shows overexpression of cadherin 1 in inclusion body myositis. Annals of neurology. 2022 Mar;91:317–28.

8. Amici DR, Pinal-Fernandez I, Mázala DAG, Lloyd TE, Corse AM, Christopher-Stine L, et al. Calcium dysregulation, functional calpainopathy, and endoplasmic reticulum stress in sporadic inclusion body myositis. Acta neuropathologica communications. 2017 Mar;5:24.

9. Amici DR, Pinal-Fernandez I, Christopher-Stine L, Mammen AL, Mendillo ML. A network of core and subtype-specific gene expression programs in myositis. Acta neuropathologica. 2021 Nov;142:887–98.

10. Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM. The sanger FASTQ file format for sequences with quality scores, and the solexa/illumina FASTQ variants. Nucleic acids research. 2010 Apr;38:1767–71.

11. Bolger AM, Lohse M, Usadel B. Trimmomatic: A flexible trimmer for illumina sequence data. Bioinformatics (Oxford, England). 2014 Aug;30:2114–20.

12. Chen S, Zhou Y, Chen Y, Gu J. Fastp: An ultra-fast all-in-one FASTQ preprocessor. Bioinformatics (Oxford, England). 2018 Sep;34:i884–90.

13. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: Ultrafast universal RNA-seq aligner. Bioinformatics (Oxford, England). 2013 Jan;29:15–21.

14. Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. Nature methods. 2017 Apr;14:417–9.

15. Frankish A, Uszczynska B, Ritchie GRS, Gonzalez JM, Pervouchine D, Petryszak R, et al. Comparison of GENCODE and RefSeq gene annotation and the impact of reference geneset on variant effect prediction. BMC genomics. 2015;16 Suppl 8:S2.

16. Zhao S, Zhang B. A comprehensive evaluation of ensembl, RefSeq, and UCSC annotations in the context of RNA-seq read mapping and gene quantification. BMC genomics. 2015 Feb;16:97.

17. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome biology. 2014;15:550.

18. Law CW, Chen Y, Shi W, Smyth GK. Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. Genome biology. 2014 Feb;15:R29.

19. Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome biology. 2010;11:R25.

20. Khatri P, Sirota M, Butte AJ. Ten years of pathway analysis: Current approaches and outstanding challenges. PLoS computational biology. 2012;8:e1002375.

21. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America. 2005 Oct;102:15545–50.

22. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, et al. Sustainable data analysis with snakemake. F1000Research. 2021;10:33.

23. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. PloS one. 2017;12:e0177459.

Apéndices

1. Código del proyecto

1.1 Demultiplexación

Makefile

```
MAKEFLAGS += -j

# Project path and folder
mkfile_path = $(abspath $(lastword $(MAKEFILE_LIST)))
project_folder = $(dir $(mkfile_path))
src = $(project_folder)src/
bcl2fastq_def = $(src)bcl2fastq-v2-20-0.def
bcl2fastq_sif = /data2/mdgroup/images/bcl2fastq-v2-20-0.sif

runs=$(wildcard $(project_folder)*/SampleSheet.csv)
all_demux=$(patsubst $(project_folder)%/SampleSheet.csv,
 ↪  $(project_folder)%/laneBarcode.html, $(runs))

.PHONY: all new rm git

all: $(all_demux)

$(bcl2fastq_sif): $(bcl2fastq_def)
   module load singularity/3.4.1; \
   singularity build --remote -f $(bcl2fastq_sif) $(bcl2fastq_def)

%/laneBarcode.html: %/bcl_path.txt %/SampleSheet.csv $(src)demux.sh
 ↪  $(bcl2fastq_sif)
   source $<; \
   qsub -sync y -o $$(dirname $@)/log.txt $(src)demux.sh $(bcl2fastq_sif)
 ↪  $${bcl_path} $$(dirname $@)
```

```
new:
   mkdir $(project_folder)$(d); \
   echo bcl_path=${b} > $(project_folder)$(d)/bcl_path.txt; \
   cp $(src)tmpSampleSheet.csv $(project_folder)${d}/SampleSheet.csv

rm:
   rm -r ./$(d)/FASTQ ./$(d)/log.txt ./$(d)/laneBarcode.html

git:
   git add .; \
   git commit -m "$m"; \
   git push;
```

demux.sh

```
#!/bin/bash

#qsub options
#$ -N bcl2fastq
#$ -r y
#$ -V
#$ -j y
#$ -pe threaded 14

#DATA TO MODIFY
BCL2FASTQ_SIF=${1}
BCL_PATH=${2}
SAMPLE_PATH=${3}

#FILE LOCALIZATION
SAMPLE_SHEET_PATH="${SAMPLE_PATH}/SampleSheet.csv"
FASTQ_PATH="${SAMPLE_PATH}/FASTQ/"

#Bcl2Fastq demultiplexing
module load singularity/3.4.1

singularity exec --bind /illumina:/illumina ${BCL2FASTQ_SIF} \
    bcl2fastq \
        --runfolder-dir ${BCL_PATH} \
        --output-dir ${FASTQ_PATH} \
        --sample-sheet ${SAMPLE_SHEET_PATH} \
        -r 4 -w 4 -p 14 \
        --barcode-mismatches 0 \
        --no-lane-splitting
```

```
rename _001.fastq.gz .fastq.gz ${FASTQ_PATH}Files/*.fastq.gz
for i in ${FASTQ_PATH}Files/*.fastq.gz; do
    arr_filename=(${i//_/ })
    mv ${i} ${i/_${arr_filename[-2]}_/_};
done

cp ${SAMPLE_PATH}/FASTQ/Reports/html/*/all/all/all/laneBarcode.html
 ↪  ${SAMPLE_PATH}/
```

1.2. Alineamiento y cuantificación

Makefile

```
MAKEFLAGS += -j200

# Project vars
debug = 0
cluster_project_folder = $(shell basename -s .git `git config --get
 ↪  remote.origin.url`)
mkfile = $(abspath $(lastword $(MAKEFILE_LIST)))
project_folder = $(dir $(mkfile))
OS = $(shell uname)
master_url = "https://github.com/musclediseasegroup/mdtools"

# Project paths
src = $(project_folder)src/
master_repo = $(src).master_repo/
data = $(project_folder)data/
results = $(project_folder)results/
fastq = $(data)fastq/
clean_fastq = $(data)clean_fastq/
salmon = $(data)salmon/
cluster_salmon_counts =
 ↪  ~/projects/$(cluster_project_folder)/data/salmon/gene_counts.csv
multiqc = $(data)multiqc/
metadata_file = $(data)metadata.csv
seq_vars_file = $(data)seq_vars.txt
local_counts = $(data)counts/
local_counts_file = $(data)counts/gene_counts.csv
tmm = $(data)tmm/
tmm_file = $(data)tmm/gene_tmm.csv
tmm_file_t = $(data)tmm/gene_tmm_t.csv
limma = $(results)limma/
```

```
# Genome variables
genome_version = GRCh38
genome_release = 39
genome_species = human
genome = $(HOME)/reference_genome/$(genome_species)/gencode/
  $(genome_version)/$(genome_release)/
genome_fasta = $(genome)genome.fa.gz
transcripts = $(genome)transcripts.fa.gz
gtf = $(genome)genes.gtf.gz
salmon_index = $(genome)salmon_index/

samples=$(shell cat $(metadata_file) | cut -d, -f1 | tail -n +2)
all_salmon=$(patsubst %, $(salmon)%/quant.sf, $(samples))

.PHONY: all clean update_src git
.PRECIOUS: $(src)%.py $(src)%.sh $(src)%.R $(fastq)%.ok $(clean_fastq)%/.ok
 ↪  $(salmon)%/.ok

ifeq ($(OS),Linux)
all: $(multiqc)multiqc_report.html ## Get the gene_counts and quality
 ↪  control in the cluster
else
all: $(local_counts_file) \
    $(tmm_file_t) \
    $(limma).ok ## Run the local part of the project
endif

$(master_repo).ok:
   @mkdir -p $(src); \
   git clone --depth 1 $(master_url) $(master_repo); \
   cd $(master_repo); \
   touch $(master_repo).ok; \
   echo "Master repo imported"

$(src)%.py $(src)%.sh $(src)%.R: | $(master_repo).ok
   @cp $(master_repo)$*.* $@; \
   echo "Master $* copied to src";

$(seq_vars_file): | $(src)get_len_fastq.py $(src)keep_r2.py
   @echo "len_fastq=$(shell python3 $(src)get_len_fastq.py)" > $@; \
   echo "keep_r2=$(shell python3 $(src)keep_r2.py)" >> $@

$(fastq)%.ok: $(metadata_file) | $(seq_vars_file) $(src)get_fastq.py
   @mkdir -p $(fastq)logs; \
   source $(seq_vars_file); \
```

```
	if [ "$(debug)" = "1" ]; then \
	   python3 $(src)get_fastq.py -s $* -k $$keep_r2 --debug; \
	   echo "$*: FASTQ files obtained in DEBUG mode (1000 first lines)"; \
	else \
	   python3 $(src)get_fastq.py -s $* -k $$keep_r2; \
	   echo "$*: FASTQ files obtained"; \
	fi; \
	ls -lah $(fastq)$*.* > $(fastq)logs/$*.log

$(clean_fastq)%/*_fastp.html: $(fastq)%.ok | $(seq_vars_file) $(src)fastp.sh
	@source $(seq_vars_file); \
	mkdir -p $(clean_fastq)logs; \
	qsub -sync y -o $(clean_fastq)logs/$*.log $(src)fastp.sh $(fastq)
	↪  $(clean_fastq) $* $$len_fastq; \
	echo "$*: Sequence cleanup finished"

$(genome).ok: | $(src)get_annotation.R
	@mkdir -p $(genome); \
	wget -P $(genome) ftp://ftp.ebi.ac.uk/pub/databases/gencode/
	 Gencode_$(genome_species)/release_$(genome_release)/
	 $(genome_version).primary_assembly.genome.fa.gz; \
	wget -P $(genome) ftp://ftp.ebi.ac.uk/pub/databases/gencode/
	 Gencode_$(genome_species)/release_$(genome_release)/
	 gencode.v$(genome_release).transcripts.fa.gz; \
	wget -P $(genome) ftp://ftp.ebi.ac.uk/pub/databases/gencode/
	 Gencode_$(genome_species)/release_$(genome_release)/
	 gencode.v$(genome_release).annotation.gtf.gz; \
	wget -P $(genome) ftp://ftp.ebi.ac.uk/pub/databases/gencode/
	 Gencode_$(genome_species)/release_$(genome_release)/
	 gencode.v$(genome_release).metadata.HGNC.gz; \
	wget -P $(genome) ftp://ftp.ebi.ac.uk/pub/databases/gencode/
	 Gencode_$(genome_species)/release_$(genome_release)/
	 gencode.v$(genome_release).metadata.EntrezGene.gz; \
	mv $(genome)*.genome.fa.gz $(genome_fasta); \
	mv $(genome)*.transcripts.fa.gz $(transcripts); \
	mv $(genome)*.gtf.gz $(gtf); \
	Rscript $(src)get_annotation.R -g $(genome); \
	touch $(genome).ok; \
	echo "Gencode genome downloaded"

$(salmon_index).ok: $(genome).ok | $(src)salmon_index.sh
	@qsub -sync y -o $(salmon_index).index_build.log $(src)salmon_index.sh
	 ↪  $(genome_fasta) $(transcripts) $(salmon_index); \
	echo "Salmon index generated"
```

```
$(salmon)%/quant.sf: $(clean_fastq)%/*_fastp.html $(salmon_index).ok |
 ↪  $(src)salmon.sh
   @mkdir -p $(salmon)logs; \
   qsub -sync y -o $(salmon)logs/$*.log $(src)salmon.sh $(salmon) $*
 ↪  $(clean_fastq) $(salmon_index); \
   echo "$*: SALMON alignment finished"

$(salmon)gene_counts.csv: $(all_salmon) | $(src)tximport.R
   @Rscript $(src)tximport.R -g $(genome); \
   echo "SALMON count aggregation finished"

$(multiqc)multiqc_report.html: $(salmon)gene_counts.csv
   @multiqc -f --interactive -o $(multiqc) $(project_folder); \
   echo "Multiqc finished running"

$(local_counts_file):
   @mkdir -p $(local_counts); \
   scp nih:$(cluster_salmon_counts) $@; \
   echo "Imported counts from cluster"

$(tmm_file_t): $(local_counts_file) | $(src)counts_to_tmm.R
 ↪  $(src)setup_tmm.R
   @mkdir -p $(tmm); \
   Rscript $(src)counts_to_tmm.R -c $< -o $(tmm_file); \
   Rscript $(src)setup_tmm.R -m $(metadata_file) -o $(tmm_file_t); \
   echo "Counts to TMM finished"

$(limma).ok: $(local_counts_file) $(metadata_file) | $(src)limma.R
   Rscript $(src)limma.R --reference_group "nt" -c $(local_counts_file) -m
 ↪  $(metadata_file) -o $(limma) -r "nt" -v group; \
   Rscript $(src)limma.R --reference_group "nt" -c $(local_counts_file) -m
 ↪  $(metadata_file) -o $(limma) -r "all" -v group; \
   touch $(limma).ok; \
   echo "Limma finished"

clean: ## Clean project
   @rm -rf $(master_repo) \
         $(fastq) \
         $(clean_fastq) \
         $(multiqc) \
         $(salmon) \
         $(seq_vars_file) \
         $(local_counts) \
         $(tmm);
   echo "Cleaned project"
```

```
update_src: ## Update scripts from central repository
	@rm -rf $(master_repo); \
	make $(master_repo).ok; \
	make $(src)*.*; \
	echo "Updated src scripts"

git: ## Update local and remote git
	@git add .; \
	git commit -m "$m"; \
	git push; \
	echo "Updated local and remote git"

help: ## Show this help
	@grep -E '^[a-zA-Z_-]+:.*?## .*$$' $(MAKEFILE_LIST) | awk 'BEGIN {FS =
	 ↪  ":.*?## "}; {printf "\033[36m%-30s\033[0m %s\n", $$1, $$2}'
```

get_len_fastq.py

```
#!/usr/bin/env python3
# *_* coding: utf-8 *_*

"""
Get minimum length of the FASTQ reads of a project
"""
__author__ = "Iago Pinal-Fernandez"
__copyright__ = "Copyright 2022, Iago Pinal-Fernandez"
__license__ = "MIT"
__version__ = "1.0.1"

import csv
import argparse
import os
import glob
import gzip

project_path = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Executable options
parser = argparse.ArgumentParser(description='Script to copy fastq files
 ↪  from a single \
                                 sample based on file information',

 ↪  formatter_class=argparse.ArgumentDefaultsHelpFormatter)
parser.add_argument("-r", "--runs", help="Sequencing runs path",
```

```
                    default=os.path.join(os.path.expanduser("~"), "runs"))
parser.add_argument("-l", "--min_length", help="Minimum length to consider",
                    default=50)
parser.add_argument("-m", "--meta", help="Metadata file",
                    default=os.path.join(project_path, "data/metadata.csv"))
parser.add_argument(
    "-f", "--fastq", help="Sample path column", default="fastq")
args = parser.parse_args()

len_lines = []
with open(args.meta, 'r') as file:
    my_reader = csv.DictReader(file, delimiter=',')
    for row in my_reader:
        fastq_info = row[args.fastq]
        for fastq in fastq_info.split(" "):
            fastq_folder = fastq.split("/")[0]
            fastq_file = fastq.split("/")[1]
            r1_fastq = glob.glob(os.path.join(
                args.runs, fastq_folder, "**", fastq_file + "_" + "*" + "R1"
 ↪  + "*" + ".fastq.gz"),
                recursive=True)[0]
            file = gzip.open(r1_fastq)
            line_n = 0
            len_fastq = 0
            while line_n < 1000:
                line = file.readline()
                if not line:
                    break
                if (line_n) % 4 == 1 and len_fastq < len(line)-1:
                    len_fastq = len(line)-1
                    # Stop if the length is equal to the minimum length
                    if (len_fastq == args.min_length):
                        break
                line_n = line_n + 1
            len_lines.append(len_fastq)
        if min(len_lines) == args.min_length:
            break

print(min(len_lines))
```

keep_r2.py

```
#!/usr/bin/env python3
# *_* coding: utf-8 *_*
```

```
"""
Script to detect if all the R1 have a corresponding R2 for a project
Returns true or false
"""
__author__ = "Iago Pinal-Fernandez"
__copyright__ = "Copyright 2022, Iago Pinal-Fernandez"
__license__ = "MIT"
__version__ = "1.0.0"

import csv
import argparse
import os
import glob
import sys

project_path = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Executable options
parser = argparse.ArgumentParser(description='Script to copy fastq files
 ↪  from a single \
                                 sample based on file information',

 ↪  formatter_class=argparse.ArgumentDefaultsHelpFormatter)
parser.add_argument("-r", "--runs", help="Sequencing runs path",
                    default=os.path.join(os.path.expanduser("~"), "runs"))
parser.add_argument("-m", "--meta", help="Metadata file",
                    default=os.path.join(project_path, "data/metadata.csv"))
parser.add_argument("-f", "--fastq", help="Sample path column",
 ↪  default="fastq")
args = parser.parse_args()

keepr2 = True
with open(args.meta, 'r') as file:
    my_reader = csv.DictReader(file, delimiter=',')
    for row in my_reader:
        fastq_info = row[args.fastq]
        for fastq in fastq_info.split(" "):
            fastq_folder = fastq.split("/")[0]
            fastq_file = fastq.split("/")[1]
            r2_fastq = glob.glob(os.path.join(
                args.runs, fastq_folder, "**", fastq_file + "_" + "*" + "R2"
 ↪  + "*" + ".fastq.gz"),
                recursive=True)
            if r2_fastq == []:
                keepr2 = False
```

```
                print(False)
                sys.exit(0)
print(True)
```

get_fastq.py

```
#!/usr/bin/env python3
# *_* coding: utf-8 *_*

"""
Script to copy fastq files from a single sample based on metadata.csv
 ↪  information
"""
__author__ = "Iago Pinal-Fernandez"
__copyright__ = "Copyright 2022, Iago Pinal-Fernandez"
__license__ = "MIT"
__version__ = "1.0.1"

import csv
import argparse
import os
import time
import glob

project_path = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Executable options
parser = argparse.ArgumentParser(description='Script to copy fastq files
 ↪  from a single \
                                 sample based on file information',

 ↪  formatter_class=argparse.ArgumentDefaultsHelpFormatter)
parser.add_argument("-s", "--sample", help="Sample name")
parser.add_argument("-r", "--runs", help="Sequencing runs path",
                    default=os.path.join(os.path.expanduser("~"), "runs"))
parser.add_argument("-m", "--meta", help="Metadata file",
                    default=os.path.join(project_path, "data/metadata.csv"))
parser.add_argument(
    "-f", "--fastq", help="Sample path column", default="fastq")
parser.add_argument(
    "-n", "--name", help="Sample name column", default="sample")
parser.add_argument("-o", "--outdir", help="Output directory",
                    default=os.path.join(project_path, "data/fastq"))
parser.add_argument("-d", "--debug", help="FAST in debug mode: get the first
 ↪  1000 lines of each file",
```

```
                    default=False, action='store_true')
parser.add_argument("-k", "--keepr2", help="Keep read 2 in paired-end
 ↪  studies",
                    default="False")
args = parser.parse_args()

if not os.path.exists(args.outdir):
    os.mkdir(args.outdir)

with open(args.meta, 'r') as file:
    my_reader = csv.DictReader(file, delimiter=',')
    for row in my_reader:
        if args.sample == row[args.name]:
            fastq_info = row[args.fastq]
            break

r1_list = []
r2_list = []
for fastq in fastq_info.split(" "):
    fastq_folder = fastq.split("/")[0]
    fastq_file = fastq.split("/")[1]
    r1_list.append(glob.glob(os.path.join(
        args.runs, fastq_folder, "**", fastq_file + "_" + "*" + "R1" + "*" +
 ↪  ".fastq.gz"),
        recursive=True)[0])
    if args.keepr2 == "True":
        r2_fastq = glob.glob(os.path.join(
            args.runs, fastq_folder, "**", fastq_file + "_" + "*" + "R2" +
 ↪  "*" + ".fastq.gz"),
            recursive=True)
        if r2_fastq != []:
            r2_list.append(r2_fastq[0])

# generate symlinks if there is only one fastq directory or new merged
 ↪  fastq files
for n, r_list in enumerate([r1_list, r2_list]):
    file_name = args.sample + "_R" + str(n+1) + ".fastq.gz"
    if len(r_list) == 0:
        continue
    # If debug mode, get first 1000 lines of file and exit loop cycle
    if args.debug:
        os.system("head " + r_list[0] + " -n 1000 > " +
                  os.path.join(args.outdir, file_name))
        break
    elif len(r_list) == 1:
```

```
        if not os.path.exists(os.path.join(args.outdir, file_name)):
            os.symlink(r_list[0], os.path.join(args.outdir, file_name))
    elif len(r_list) > 1:
        dst_file = os.path.join(args.outdir, file_name)
        tmp_file = os.path.join(args.outdir, file_name + ".tmp")
        open(tmp_file, 'w').close()
        qsub_script = "echo 'cat " + " ".join(r_list) + " > " + dst_file +
↪   ";" + \
            "rm " + tmp_file + ";' | " + \
            "qsub -j y -o /dev/null -N get_fastq"
        os.system(qsub_script)
        while os.path.exists(tmp_file):
            time.sleep(1)

open(os.path.join(args.outdir, args.sample + ".ok"), 'w').close()
```

fastp.sh

```
#!/bin/bash

# Quality control with fastp
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 1.0.0

#qsub options
#$ -N fastp
#$ -r y
#$ -V
#$ -j y
#$ -pe threaded 2

#Tag script variables
FASTQ_PATH=${1}
CLEAN_FASTQ_PATH=${2}
SAMPLE=${3}
LEN_FASTQ=${4}

echo $SAMPLE

mkdir ${CLEAN_FASTQ_PATH}${SAMPLE}

if [ -f ${FASTQ_PATH}${SAMPLE}_R2.fastq.gz ]
then
    fastp -i ${FASTQ_PATH}${SAMPLE}_R1.fastq.gz \
```

```
        -I ${FASTQ_PATH}${SAMPLE}_R2.fastq.gz \
        -o ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_R1.fastq.gz  \
        -O ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_R2.fastq.gz \
        -h ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_fastp.html \
        -j ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_fastp.json \
        --max_len1=${LEN_FASTQ}
else
    fastp -i ${FASTQ_PATH}${SAMPLE}_R1.fastq.gz \
        -o ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_R1.fastq.gz  \
        -h ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_fastp.html \
        -j ${CLEAN_FASTQ_PATH}${SAMPLE}/${SAMPLE}_fastp.json \
        --max_len1=${LEN_FASTQ}
fi

rm ${FASTQ_PATH}${SAMPLE}_R*.fastq.gz

touch ${CLEAN_FASTQ_PATH}${SAMPLE}/.ok

echo "FASTQ files ${SAMPLE} cleaned"
```

get_annotation.R

```
#!/usr/bin/env Rscript

# Get ensembl annotation
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 2.0.0

pacman::p_load(optparse, tximport, GenomicFeatures)

option_list <- list(
  make_option(c("-g", "--genome"),
              default="~/reference_genome/human/gencode/GRCh38/39/",
              help="Genome path")
)

opt <- parse_args(OptionParser(option_list=option_list))

# Get mapping of transcript to gene ID
reference_files <- list.files(opt$genome)
gtf_file <- paste0(opt$genome, reference_files[grepl("gtf",
 ↪  reference_files)])
txdb <- makeTxDbFromGFF(file = gtf_file)
k <- keys(txdb, keytype = "TXNAME")
```

```
tx2gene <- select(txdb, k, "GENEID", "TXNAME")
write.csv(tx2gene, file.path(opt$genome, "tx2gene.csv"), row.names=FALSE,
 ↪  quote=F)

# Get HGNC symbols and ENTREZ IDs
get_ref <- function(x) {
    file <- reference_files[grepl(x, reference_files)]
    temp <- read.table(paste0(opt$genome, file), sep="\t", header=FALSE)
    temp
}
hgnc <- get_ref("HGNC")
colnames(hgnc) <- c("TXNAME", "SYMBOL", "HGNCID")
entrezgene <- get_ref("EntrezGene")
colnames(entrezgene) <- c("TXNAME", "ENTREZGENE")

tx2gene_merged <- merge(tx2gene, hgnc, by="TXNAME", all.x=TRUE)
tx2gene_merged <- merge(tx2gene_merged, entrezgene, by="TXNAME", all.x=TRUE)
tx2gene_merged <- tx2gene_merged[!duplicated(tx2gene_merged$GENEID), -1]
write.csv(tx2gene_merged, file.path(opt$genome, "geneid_hgnc_entrez.csv"),
 ↪  row.names=FALSE, quote=F)
```

salmon_index.sh

```
#!/bin/bash

# Generate salmon index
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 1.0.0

#qsub options
#$ -N salmon_index
#$ -r y
#$ -V
#$ -j y
#$ -pe threaded 16

GENOME=${1}
TRANSCRIPTS=${2}
SALMON_INDEX=${3}

cd ${SALMON_INDEX}

grep "^>" <(gunzip -c ${GENOME}) | cut -d " " -f 1 > decoys.txt
sed -i.bak -e 's/>//g' decoys.txt
```

```
cat ${TRANSCRIPTS} ${GENOME} > gentrome.fa.gz

salmon index i \
    -t gentrome.fa.gz \
    -d decoys.txt \
    -p 16 \
    -i ${SALMON_INDEX} \
    --gencode

touch ${SALMON_INDEX}.ok
```

salmon.sh

```
#!/bin/bash

# Salmon quantification
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 1.0.0

#qsub options
#$ -N salmon
#$ -r y
#$ -V
#$ -j y
#$ -pe threaded 8

#Tag script variables
ALIGNED_SALMON_PATH=${1}
SAMPLE=${2}
FASTQ_PATH=${3}
SALMON_INDEX=${4}
THREADS=8

#Alignment using STAR
mkdir ${ALIGNED_SALMON_PATH}${SAMPLE}

if [ -f ${FASTQ_PATH}${SAMPLE}/${SAMPLE}_R2.fastq.gz ]
then
    salmon quant \
        -i ${SALMON_INDEX} \
        -l A \
        -1 ${FASTQ_PATH}${SAMPLE}/${SAMPLE}_R1.fastq.gz \
        -2 ${FASTQ_PATH}${SAMPLE}/${SAMPLE}_R2.fastq.gz \
```

```
        -p ${THREADS} \
        --validateMappings \
        -o ${ALIGNED_SALMON_PATH}${SAMPLE}
else
    salmon quant \
        -i ${SALMON_INDEX} \
        -l A \
        -r ${FASTQ_PATH}${SAMPLE}/${SAMPLE}_R1.fastq.gz \
        -p ${THREADS} \
        --validateMappings \
        -o ${ALIGNED_SALMON_PATH}${SAMPLE}
fi

#Delete temporary files and tag
rm ${FASTQ_PATH}${SAMPLE}/${SAMPLE}_R*.fastq.gz

touch ${ALIGNED_SALMON_PATH}${SAMPLE}/.ok
```

tximport.R

```
#!/usr/bin/env Rscript

# Get salmon gene counts matrix
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 2.0.2

pacman::p_load(optparse, tximport, GenomicFeatures)

option_list <- list(
  make_option(c("-s", "--salmon"),
              default="./data/salmon/",
              help="Salmon path"),
  make_option(c("-m", "--metadata"),
              default="./data/metadata.csv",
              help="Metadata file"),
  make_option(c("-i", "--id"),
              default="sample",
              help="Variable with the sample ids"),
  make_option(c("-o", "--output_path"),
              default="./data/salmon/",
              help="Output path"),
  make_option(c("-g", "--genome"),
              default="~/reference_genome/human/gencode/GRCh38/39/",
              help="Genome path")
```

```
)

opt <- parse_args(OptionParser(option_list=option_list))

dir.create(opt$output_path,
             showWarnings = FALSE,
             recursive = TRUE)

samples <- read.csv(opt$metadata)[[opt$id]]
files <- file.path(opt$salmon, samples, "quant.sf")
names(files) <- samples

# Get mapping of transcript to gene ID
tx2gene <- read.csv(file.path(opt$genome, "tx2gene.csv"))
tx2gene_merged <- read.csv(file.path(opt$genome, "geneid_hgnc_entrez.csv"))

txi_salmon <- tximport(files, type = "salmon", tx2gene = tx2gene)
txi_salmon$counts <- cbind(gene_id = rownames(txi_salmon$counts),
 ↪  txi_salmon$counts)

# Translate ENSEMBL ID to gene symbol
counts <- merge(txi_salmon$counts, tx2gene_merged, by.x="gene_id",
 ↪  by.y="GENEID", all.x=TRUE)
counts$gene_id <- toupper(ifelse(is.na(counts$SYMBOL), counts$gene_id,
 ↪  counts$SYMBOL))
counts <- counts[!(names(counts) %in% c("SYMBOL", "HGNCID", "ENTREZGENE"))]

write.csv(counts, file.path(opt$output_path, "gene_counts.csv"),
 ↪  row.names=FALSE, quote=F)
```

counts_to_tmm.R

```
#!/usr/bin/env Rscript

# Obtain TMM normalization from raw counts
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 1.0.0

pacman::p_load(optparse, edgeR)

option_list <- list(
  make_option(c("-c", "--counts"),
              default="./data/counts/gene_counts.csv",
              help="Gene counts file"),
```

```
  make_option(c("-o", "--output_path"),
              default="./data/tmm/gene_tmm.csv",
              help="Output path")
)

opt <- parse_args(OptionParser(option_list=option_list))

countData <- read.csv(opt$counts)
dge <- DGEList(data.frame(countData, row.names=TRUE))
dge <- calcNormFactors(dge, method = "TMM")
tmm <- cpm(dge)
tmm <- cbind(gene_id = rownames(tmm), tmm)
rownames(tmm) <- NULL

write.csv(tmm, opt$output_path, row.names = FALSE)
```

setup_tmm.R

```
#!/usr/bin/env Rscript

# Setup TMM file for visualization purposes
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 1.0.1

pacman::p_load(optparse, data.table)

option_list <- list(
  make_option(c("-t", "--tmm"),
              default="./data/tmm/gene_tmm.csv",
              help="TMM file"),
  make_option(c("-g", "--gene_id"),
              default="gene_id",
              help="Variable with the gene ids"),
  make_option(c("-m", "--metadata"),
              default="./data/metadata_tmp.csv",
              help="Metadata file"),
  make_option(c("-i", "--sample_id"),
              default="sample",
              help="Variable with the sample ids in the metadata file"),
  make_option(c("-o", "--output_file"),
              default="./data/tmm/gene_tmm_t.csv",
              help="Output file")
)
```

```
opt <- parse_args(OptionParser(option_list=option_list))

tmm <- read.csv(opt$tmm)
metadata <- read.csv(opt$metadata)

tmm_t <- data.table::transpose(tmm, keep.names=opt$sample_id,
 ↪  make.names=opt$gene_id)
merged_tmm_t <- merge(tmm_t, metadata, by=opt$sample_id)

write.csv(merged_tmm_t, opt$output_file, row.names=FALSE)
```

1.3. Expresión diferencial, análisis de pathways y análisis gráfico

limma.R

```
#!/usr/bin/env Rscript

# Differential expression with limma
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 2.0.0

pacman::p_load(optparse, limma, edgeR, tidyverse, foreach, doParallel,
 ↪  parallel, dplyr)

option_list <- list(
  make_option(c("-c", "--counts"),
    default = "./data/counts/gene_counts.csv",
    help = "Gene counts file"
  ),
  make_option(c("-m", "--metadata"),
    default = "./data/metadata_tmp.csv",
    help = "Metadata file"
  ),
  make_option(c("-i", "--id"),
    default = "sample",
    help = "Variable with the sample ids"
  ),
  make_option(c("-v", "--variable"),
    default = "group",
    help = "Variable to do the differential expression"
  ),
  make_option(c("-g", "--reference_group"),
    default = "nt",
    help = "Reference group"
```

```
  ),
  make_option(c("-o", "--output_path"),
    default = "./results/limma/",
    help = "Output path"
  ),
  make_option(c("-f", "--filter"),
    default = NULL,
    help = "Dataset filter"
  ),
  make_option(c("-t", "--adjust_terms"),
    default = NULL,
    help = "Covariates for multivariate analysis"
  ),
  make_option(c("-r", "--restrict"),
    default = "",
    help = "Restrict differential expression to regex"
  )
)

opt <- parse_args(OptionParser(option_list = option_list))

diff_exp <- function(srch_grp1, srch_grp2, counts, metadata, opt) {
  if ((srch_grp1 == opt$reference_group | srch_grp1 == "all") & srch_grp2
   ↪  != "all") {
    tmp <- srch_grp1
    srch_grp1 <- srch_grp2
    srch_grp2 <- tmp
  }

  cat("Starting", srch_grp1, "vs.", srch_grp2, "\n")

  if (srch_grp2 == "all") {
    tmpColData <- metadata %>%
      filter(tmp == srch_grp1 | tmp != srch_grp1) %>%
      mutate(tmp = ifelse(tmp == srch_grp1, srch_grp1, srch_grp2))
  } else {
    tmpColData <- metadata %>%
      filter(tmp == srch_grp1 | tmp == srch_grp2)
  }

  tmpcount_data <- counts[, c("gene_id", tmpColData[[opt$id]])]

  # Build DGEList object
  d0 <- DGEList(data.frame(tmpcount_data, row.names = TRUE))
```

```
# Calculate normalization factors
d0 <- calcNormFactors(d0)

# Filter low-expressed genes
keep.exprs <- filterByExpr(d0, group = tmpColData$tmp)
d <- d0[keep.exprs, , keep.lib.sizes = FALSE]

# Fit model
if (is.null(opt$adjust_terms)) {
  design <- model.matrix(~ 0 + tmp, tmpColData)
} else {
  formula <- paste0("~0 + tmp + ", opt$adjust_terms)
  design <- model.matrix(as.formula(formula), tmpColData)
}

# Voom
y <- voom(d, design, plot = F)

# Fit linear model
fit <- lmFit(y, design)

contr <- makeContrasts(
  contrasts = paste0(
    "tmp", srch_grp1,
    "-tmp", srch_grp2
  ),
  levels = design
)

tmp <- contrasts.fit(fit, contr)

tmp <- eBayes(tmp)

res <- topTable(tmp, sort.by = c("P"), n = Inf)

# Store results
dir.create(opt$output_path,
  showWarnings = FALSE,
  recursive = TRUE
)

res_file <- paste0(opt$output_path, srch_grp1, "_", srch_grp2, ".csv")

write.csv(as.data.frame(res),
  file = res_file,
```

```
    quote = FALSE,
    row.names = TRUE
  )

  cat("Finished", srch_grp1, "vs.", srch_grp2, "\n")
}

# Import gene counts
counts <- read_csv(opt$counts)
metadata <- read_csv(opt$metadata)

if (!is.null(opt$filter)) {
  metadata <- metadata %>%
    filter(rlang::eval_tidy(rlang::parse_expr(opt$filter)))
  filtered_samples <- metadata[[opt$id]]
  counts <- counts[, c("gene_id", filtered_samples)]
}

metadata$tmp <- metadata[[opt$variable]]

comparison_groups <- unique(metadata$tmp)

if (length(comparison_groups) == 2) {
  srch_grp1 <- comparison_groups[1]
  srch_grp2 <- comparison_groups[2]

  diff_exp(srch_grp1, srch_grp2, counts, metadata, opt)
} else {
  combinations <- combn(c(comparison_groups, "all"), 2)

  cl <- makeCluster(detectCores(), outfile = "")
  registerDoParallel(cl)

  # Loop over all the combinations of the variable
  foreach(
    n = 1:(length(combinations) / 2),
    .verbose = T,
    .packages = c("tidyverse", "edgeR", "limma")
  ) %dopar% {
    srch_grp1 <- combinations[1, n]
    srch_grp2 <- combinations[2, n]

    if (grepl(opt$restrict, srch_grp1) | grepl(opt$restrict, srch_grp2)) {
      diff_exp(srch_grp1, srch_grp2, counts, metadata, opt)
    }
```

```
  }

  stopCluster(cl)
}
```

cluster_profiler.R

```
#!/usr/bin/env Rscript

# Pathway analysis with cluster profiler
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
# Version: 1.0.1

pacman::p_load(optparse, clusterProfiler, ggplot2, ReactomePA, DOSE,
 ↪  enrichplot, SBGNview, biomaRt, org.Hs.eg.db)

option_list <- list(
  make_option(c("-d", "--diff_exp"),
    default = "./results/limma/mi2_all.csv",
    help = "Differential expression file"
  ),
  make_option(c("-i", "--id"),
    default = "X",
    help = "Variable with the sample ids"
  ),
  make_option(c("-f", "--fc"),
    default = "logFC",
    help = "Variable with the fold-change"
  ),
  make_option(c("-o", "--output_path"),
    default = "./results/cluster_profiler/",
    help = "Output path"
  )
)

opt <- parse_args(OptionParser(option_list = option_list))

dir.create(opt$output_path, recursive = T, showWarnings = F)

df <- read.csv(opt$diff_exp)

# Get annotation
mart <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
```

```
# Biomart
gene_list <- getBM(attributes = c("ensembl_gene_id", "hgnc_symbol",
 ↪  "gene_biotype"),
                   mart = mart,
                   uniqueRows = TRUE)

# org.Hs.eg.db
ncbi_geneids <- bitr(gene_list$ensembl_gene_id, fromType = "ENSEMBL",
                     toType = c("SYMBOL", "ENTREZID"),
                     OrgDb = org.Hs.eg.db,
                     drop = F)

gene_list <- merge(gene_list, ncbi_geneids, by.x='ensembl_gene_id',
 ↪  by.y='ENSEMBL')
gene_list["hgnc_symbol"][gene_list["hgnc_symbol"] == ""] <-
 ↪  gene_list$SYMBOL[gene_list["hgnc_symbol"] == ""]
gene_list$SYMBOL <- NULL

gene_list <- gene_list[!duplicated(gene_list$ensembl_gene_id),]
gene_list["hgnc_symbol"][is.na(gene_list[["hgnc_symbol"]]),] <-
 ↪  gene_list$ensembl_gene_id[is.na(gene_list["hgnc_symbol"])]


# Continue
df <- merge(df, gene_list, by.x = opt$id, by.y = "hgnc_symbol")
df <- df[!duplicated(df["ENTREZID"]), ]
df <- df[!is.na(df["ENTREZID"]), ]

genes <- df[[opt$fc]]
names(genes) <- df[["ENTREZID"]]
genes <- sort(genes, decreasing = T)

kk2 <- gsePathway(
  geneList = genes,
  pAdjustMethod = "BH",
  verbose = T,
  seed = T
)

write.csv(kk2, paste0(opt$output_path, "pathway_analysis.csv"))

dotplot <- dotplot(kk2,
  x = "NES",
  showCategory = 20,
  color = "p.adjust",
```

```
  font.size = 9,
  label_format = 50
)

ggsave(paste0(opt$output_path, "dotplot.pdf"), plot = dotplot)

file.create(paste0(opt$output_path, ".ok"))
```

rnaseq_tools.R

```
#!/usr/bin/env Rscript

# Tools for RNAseq
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022

pacman::p_load(ComplexHeatmap, ggplot2)

# Function to generate individual plots
individual_plot <- function(df=df, group='group', gene='ISG15',
 ↪  ylab="log2(TMM+1)", title=NULL, color=NULL) {
  ggplot(df, aes_string(x=as.name(group), y=as.name(gene),
   ↪  color=as.name(color))) +
    ggtitle(title) +
    ylab(ylab) +
    xlab("") +
    geom_boxplot(fill='grey') +
    geom_beeswarm(corral = "gutter", dodge.width = 0.75) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
          text = element_text(size = 16))
}

# Function to generate z-score heatmaps
heatmap_z_score <- function(df=df,
                            group='group',
                            gene_list=c('ISG15', 'MX1'),
                            gene_groups=NULL,
                            gene_groups_name='Gene list',
                            cluster_rows=TRUE,
                            break_range=1.5,
                            output_path='../results/heatmaps/gene_list.pdf')
                             ↪  {

  filtered_df <- data.matrix(df[, ..gene_list])
```

```
rownames(filtered_df) <- rownames(df)

z_score_df <- scale(filtered_df)
z_score_df[is.na(z_score_df)] <- 0
z_score_df <- aggregate(z_score_df, list(df[,group]), FUN=mean)
rownames(z_score_df) <- z_score_df$Group.1
z_score_df$Group.1 <- NULL

# Data frame with annotations.
mat_row <- data.frame(group = rownames(z_score_df))
rownames(mat_row) <- rownames(z_score_df)

parameter_list <- list(mat = t(z_score_df),
                       show_rownames = TRUE,
                       show_colnames = FALSE,
                       annotation_legend = TRUE,
                       annotation_names_row = FALSE,
                       cluster_rows = cluster_rows,
                       cluster_row_slices = FALSE,
                       cluster_cols = FALSE,
                       breaks = seq(-break_range, break_range, length.out =
                        ↪  100),
                       show_row_dend = FALSE,
                       show_column_dend = FALSE,
                       row_title = NULL,
                       column_split = 1:length(rownames(z_score_df)),
                       column_title = NULL,
                       top_annotation = HeatmapAnnotation(
                         foo = anno_block(labels = rownames(z_score_df))),
                       heatmap_legend_param = list(
                         title="z-score"))

if (is.null(gene_groups)) {
  parameter_list <- c(parameter_list, list(row_split = NULL,
↪  annotation_row = NA))
} else {
  mat_col <- data.frame(gene_groups)
  colnames(mat_col) <- gene_groups_name
  rownames(mat_col) <- rownames(mat_col)

  parameter_list <- c(parameter_list, list(row_split = mat_col,
↪  annotation_row = mat_col))
}

pdf(file=output_path, width=10, height=10)
```

```
    dheatmap <- do.call(ComplexHeatmap::pheatmap, parameter_list)
    draw(dheatmap)
  dev.off()
}

# Download hugo dataset
download_hugo <- function(hugo_path) {
  if (!file.exists(hugo_path)) {
    download.file('http://ftp.ebi.ac.uk/pub/databases/genenames/
      hgnc/tsv/hgnc_complete_set.txt', hugo_path)
  }

  read_delim(hugo_path)
}

# Get gene symbols from a HUGO family
hugo_gene_group <- function(hugo_df=hugo_df, gene) {
  gene_group_ids <- lapply(hugo_df$gene_group_id, strsplit, "|", fixed = T)
  matched_group <- lapply(gene_group_ids, function(x) {
    match(gene, unlist(x), nomatch = F)
  })
  matched_group <- unlist(matched_group) > 0
  gene_list <- hugo_df[matched_group, ]$symbol
  gene_list
}
```

rnaseq_tools.py

```
#!/usr/bin/env python3
# *_* coding: utf-8 *_*

"""
Tools to manage and graph RNAseq data
"""
__author__ = "Iago Pinal-Fernandez"
__copyright__ = "Copyright 2021, Iago Pinal-Fernandez"
__license__ = "MIT"

import os
import pandas as pd
import numpy as np
import seaborn as sns
import math
import matplotlib.pyplot as plt
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

```
from scipy import stats


def whiskerplot_gene(df,
                     grouping_var,
                     group_order,
                     gene,
                     fig_path=None,
                     style="boxplot",
                     ax=None,
                     ylabel=None,
                     ylim=None):
    '''
        Function to plot a whiskerplot of log2 FPKM + 1 of one gene over
↪   different categories.

        Parameters:
            df: Dataframe
            grouping_var: Grouping variable
            group_order: Order or the grouping variable
            gene: Gene
            fig_path: Path of the file to save the figure
            style: barplot, boxplot or tukey
    '''
    sns.set_style("whitegrid")
    df_subset = df[[grouping_var, gene]].copy()
    df_subset[gene] = np.log2(df_subset[gene].astype('float64') + 1)
    if style == "barplot":
        fig = sns.barplot(x=grouping_var, y=gene,
                          data=df_subset, order=group_order, ax=ax)
        fig = sns.swarmplot(x=grouping_var, y=gene,
                            data=df_subset, color="k", order=group_order,
↪   ax=ax)
    elif style == "boxplot":
        fig = sns.boxplot(x=grouping_var, y=gene, data=df_subset,
                          order=group_order, color="silver", ax=ax)
        fig = sns.swarmplot(x=grouping_var, y=gene,
                            data=df_subset, color="k", order=group_order,
↪   ax=ax)
    elif style == "tukey":
        tukey = pairwise_tukeyhsd(endog=df_subset[gene],  # Data
                                  groups=df_subset[grouping_var],  # Groups
                                  alpha=0.05)  # Significance level

        tukey._simultaneous_ci()
```

```
        sns.despine(left=True)
        fig = sns.swarmplot(x=grouping_var, y=gene, data=df_subset,
↪   color="k")
        fig.errorbar(range(len(df_subset[grouping_var].unique())),
                     df_subset[gene].groupby(
                         df_subset[grouping_var]).mean().values,
                     yerr=tukey.halfwidths,
                     linestyle='None',
                     elinewidth=2,
                     marker='')
    elif style == "line":
        fig = sns.pointplot(x=grouping_var, y=gene, data=df_subset, ci=95,
↪   markers="None", order=group_order,
                            ax=ax)  # scale = 2.0, size=3, aspect=1.5,
        # fig = sns.swarmplot(x=grouping_var, y=gene, data=df_subset,
         ↪  color="k", order=group_order, ax=ax

    # fig.ylabel('log$_{2}$(FPKM+1)')
    fig.set_ylabel(r"$\bf{" + ylabel + "}$", rotation=90)
    fig.set_xlabel('')
    fig.set_title('')
    fig.set_ylim(ylim)
    fig.spines['top'].set_visible(False)
    fig.set_xticklabels(fig.get_xticklabels(), rotation=45)
    # ax.set(ylabel=(ylabel, rotation=0), xlabel='', title='', yticks=[0,
     ↪  2,4,6], ylim=(-0.5, 6.5))
    # fig.tick_params(axis='both', which='major', labelsize=20)
    # fig.set_xticklabels(group_order)
    if fig_path:
        plt.savefig(fig_path, bbox_inches='tight')
    return fig


def whiskerplot_group(df, gene_list, group, hue_order=None, fig_path=None,
↪   ax=None, ylabel="", color="k", shift=0):
    '''
        Function to plot a whiskerplot of log2 FPKM + 1 of several genes
↪   over one clinical group.

        Parameters:
            df: Dataframe
            gene_list: List of genes
            group: List containing grouping variable and value
            fig_path: Path of the file to save the figure
            ax: axis
```

```
            ylabel: y label
    '''
    sns.set_style("whitegrid")
    df_subset = pd.DataFrame(df[gene_list].stack()
                             ).rename_axis(['sample', 'gene'])
    df_subset.columns = ['value']
    df_subset['value'] = np.log2(df_subset['value'] + 1)
    df_subset.reset_index(inplace=True)
    df_subset = df_subset.merge(pd.DataFrame(
        df[group]), left_on='sample', right_index=True)

    fig = sns.pointplot(x='gene', y='value', hue=group, hue_order=hue_order,
↪   dodge=0.5, data=df_subset, ci=95,
                        markers=".", join=False)

    fig.set_ylabel(ylabel)
    fig.set_xlabel('')
    fig.set_title('')
    fig.spines['top'].set_visible(False)
    fig.grid(which='major', axis='y', linestyle='')
    fig.xaxis.grid(True, which='minor')
    fig.set_xticks([x + 0.5 for x in list(range(len(gene_list)))],
↪   minor=True)
    fig.legend(bbox_to_anchor=(1.00, 1), loc=2, borderaxespad=0.)
    plt.xticks(rotation=45)

    # fig.tick_params(axis='both', which='major', labelsize=20)
    # fig.set_xticklabels(group_order)
    if fig_path:
        plt.savefig(fig_path, bbox_inches='tight')
    return fig


def heatmap_genes_by_group(df, grouping_var, group_order, genes, vmin=None,
↪   vmax=None, fig_path=None):
    '''
        Function to plot a heatmap of fold-change of the categories of a
↪   group
        over a set of genes.

        Parameters:
            df: Dataframe
            grouping_var: Grouping variable
            group_order: Order or the grouping variable
            genes: Set of genes
```

```
            vmin: Minimum fold-change
            vmax: Maximum fold-change
            fig_path: Path of the file to save the figure
    '''
    df_subset = df[genes + [grouping_var]]

    df_subset = df_subset.groupby(grouping_var).mean()
    df_subset = df_subset.reindex(group_order)

    for gene in genes:
        df_subset[gene] = df_subset[gene]
        mean_nt = df_subset.loc[df_subset.index == 'NT'][gene][0]
        df_subset[gene] = np.log2(df_subset[gene] / mean_nt)

    ax = sns.heatmap(df_subset, linewidths=0.5,
                     cmap="RdBu_r", vmin=vmin, vmax=vmax)
    plt.xticks(rotation=45)
    plt.ylabel('Patient Group')
    plt.title('Log$_{2}$ fold-change vs. normal')
    if fig_path:
        plt.savefig(fig_path, bbox_inches='tight')
    plt.show()
    plt.close()


def heatmap_genes_longitudinal(df, longitudinal_var, genes, vmin=None,
 ↪  vmax=None, fig_path=None):
    '''
        Function to plot a heatmap of fold-change of a gene
 ↪  longitudinally.

        Parameters:
            df: Dataframe
            longitudinal_var: Variable with the longitudinal information
            genes: Set of genes
            vmin: Minimum fold-change
            vmax: Maximum fold-change
            fig_path: Path of the file to save the figure
    '''
    df[longitudinal_var] = df[longitudinal_var].astype(int)

    df_subset = df[genes + [longitudinal_var]]

    df_subset = df_subset.groupby(longitudinal_var).mean()
```

```
    for gene in genes:
        mean_nt = df_subset.loc[df_subset.index == 0][gene][0]
        df_subset[gene] = np.log2(df_subset[gene] / mean_nt)

    df_subset = df_subset.transpose()

    ax = sns.heatmap(df_subset, linewidths=0.5,
                     cmap="RdBu_r", vmin=vmin, vmax=vmax)
    # plt.xticks(rotation=45)
    plt.ylabel('Gene')
    plt.title('Log$_{2}$ fold-change vs. Day0')
    if fig_path:
        plt.savefig(fig_path, bbox_inches='tight')
    plt.show()
    plt.close()


def heatmap_genes_correlation(df, vertical_gene_set, horizontal_gene_set,
 ↪  fig_path=None, vmin=-1, vmax=1,
                               vertical_log=True):
    '''
        Function to plot a heatmap of Spearman correlations of two gene
 ↪  sets.

        Parameters:
            df: Dataframe
            vertical_gene_set: Variable with the vertical set of genes
            horizontal_gene_set: Variable with the horizontal set of genes
            fig_path: Path of the file to save the figure
            vmin: Minimum fold-change
            vmax: Maximum fold-change
    '''
    corr_matrix = [[0] * len(horizontal_gene_set)
                   for _ in range(len(vertical_gene_set))]

    df_subset = df[list(set(horizontal_gene_set +
 ↪  vertical_gene_set))].copy()

    for gene in df_subset.columns:
        if vertical_log == False:
            if gene in vertical_gene_set:
                df_subset[gene] = np.log2(df_subset[gene] + 1)
        else:
            df_subset[gene] = np.log2(df_subset[gene] + 1)
```

```
    for i, gene_row in enumerate(vertical_gene_set):
        for j, gene_col in enumerate(horizontal_gene_set):
            corr_matrix[i][j] = df_subset[gene_row].corr(
                df_subset[gene_col], method='spearman')

    df_subset = pd.DataFrame(
        corr_matrix, index=vertical_gene_set, columns=horizontal_gene_set)

    ax = sns.heatmap(df_subset, linewidths=0.5,
                     cmap="RdBu_r", vmin=vmin, vmax=vmax)
    plt.xticks(rotation=45)
    plt.title('Spearman correlation')
    if fig_path:
        plt.savefig(fig_path, bbox_inches='tight')
    plt.show()
    plt.close()


def bivariate_scatter(df, gene1, gene2, fig_path=None, label=None,
 ↪  log_gene1=True, log_gene2=True):
    sns.set_style("white")
    if log_gene1 == True:
        gene1_value = np.log2(df[gene1] + 1)
    else:
        gene1_value = df[gene1]
    if log_gene2 == True:
        gene2_value = np.log2(df[gene2] + 1)
    else:
        gene2_value = df[gene2]
    plt.scatter(gene1_value, gene2_value, label=label)
    plt.title('Log$_{2}$ FPKM of ' + gene1 + " vs. " + gene2)
    plt.legend()
    plt.xlabel(gene1, size=20)
    plt.ylabel(gene2, size=20)
    if fig_path:
        plt.savefig(fig_path, bbox_inches='tight')
    return plt


def compact_flatten_hugo_list(path_hugo_file, filter_criteria):
    '''
    Function to flatten and filter the full list of gene names from HUGO
 ↪  by gene family

    Parameters:
```

```
        path_hugo_file: path of the file including the gene lists from
↪   HUGO
        filter_criteria: criteria to filter using the gene family
    '''
    gene_list_hugo = pd.read_csv(path_hugo_file, sep='\t')

    filtered_list =
↪   gene_list_hugo[gene_list_hugo['gene_group'].str.contains(
        filter_criteria, na=False)]
    filtered_list = filtered_list[filtered_list['status'] == 'Approved']

    return list(filtered_list['symbol'])


def filter_limma(path_input_results, path_output_results, gene_list,
↪   significant=True):
    '''
    Function to filter the limma results table based on a list of genes

    Parameters:
        path_input_results: input file
        path_output_results: output file
        gene_list: gene list filter
        significant: keep only significant genes
    '''
    df = pd.read_csv(path_input_results)
    df = df[df['gene_id'].isin(gene_list)]
    if significant == True:
        df = df[df['adj.P.Val'] < 0.05]
    outdir = os.path.dirname(path_output_results)
    if not os.path.exists(outdir):
        os.makedirs(outdir, exist_ok=True)
    df.to_csv(path_output_results, index=False)


def top_genes_limma(group_list, comparison_type, path_input_folder,
↪   path_output_table, sign_values='all',
                    number_top_genes=10):
    '''
    Function to create a table of expression with 10 most significant
↪   genes

    Parameters:
        group_list: list of groups to include in the table
        comparison_type: vs NT or vs all
```

```
        path_input_folder: path of input folder
        path_output_table: path of output table
        sign_values: filter for 'pos' or 'neg' values
        number_top_genes: number of genes from the top to retain
    Returns: Dictionary including the gene symbol for each group
    '''
    full_df = pd.DataFrame()
    genes_by_group = {}
    for group in group_list:
        if not (group == 'nt' and comparison_type == 'nt'):
            df = pd.read_csv(os.path.join(path_input_folder,
                             group + '_' + comparison_type + '.csv'))
            if sign_values == 'pos':
                df = df[df['logFC'] > 0]
            elif sign_values == 'neg':
                df = df[df['logFC'] < 0]
            df = df[['gene_id', 'logFC', 'adj.P.Val']][:number_top_genes]
            df.reset_index(drop=True, inplace=True)
            df['logFC'] = df['logFC'].round(0).astype(int)
            genes_by_group[group] = df['gene_id']
            df.columns = group + '_' + df.columns
            full_df = pd.concat([full_df, df], axis=1)
    if path_output_table != None:
        full_df.to_csv(path_output_table + ".csv", na_rep='',
                       float_format='%.2E', index=False)
    return genes_by_group


def list_genes_limma(group_list, gene_list, comparison_type='nt',
 ↪  path_input_folder='./results/tables/limma/', path_output_table='None',
                     sign_values='all'):
    '''
    Function to create a table of expression a set of genes

    Parameters:
        group_list: list of groups to include in the table
        gene_list: list of genes to include in the table
        comparison_type: vs nt or vs all
        path_input_folder: path of input folder
        path_output_table: name of output table file
        sign_values: get 'all' values or filter for 'pos' or 'neg' values
    Returns: Dictionary including the gene symbol for each group
    '''
    full_df = None
    for group in group_list:
```

```
        if not (group == 'nt' and comparison_type == 'nt'):
            df = pd.read_csv(os.path.join(path_input_folder,
                            group + '_' + comparison_type + '.csv'))
            print(df)
            df = df[df['gene_id'].isin(gene_list)]
            df = df[['gene_id', 'logFC', 'adj.P.Val']]
            if sign_values == 'pos':
                df = df[df['logFC'] > 0]
            elif sign_values == 'neg':
                df = df[df['logFC'] < 0]
            df.reset_index(drop=True, inplace=True)
            df['logFC'] = df['logFC'].round(1)
            genes_to_add = set(gene_list) - set(df['gene_id'])
            for i in genes_to_add:
                gene_to_add = pd.Series([i, 'NA', 'NA'], index=df.columns)
                df = df.append(gene_to_add, ignore_index=True)
            df = df.sort_values('gene_id')
            df = df.rename(
                columns={'logFC': 'logFC_' + group, 'adj.P.Val':
↪   'adj.P.Val_' + group})
            if full_df is None:
                full_df = df
            else:
                full_df = full_df.merge(df, on='gene_id')
    if path_output_table is not None:
        outdir = os.path.dirname(path_output_table)
        if not os.path.exists(outdir):
            os.makedirs(outdir, exist_ok=True)
        full_df.to_csv(path_output_table, na_rep='',
                       float_format='%.2E', index=False)
    return full_df


def top_genes_per_group_graph(df, genes_per_group, group_order,
↪   output_file):
    '''
    Function to create a graph of expression with 10 most significant
↪   genes

    Parameters:
        df: dataframe with FPKM values
        genes_per_group: dictionary of genes for each group
        group_order: order of groups for subgraph
        output_file: path of output file
    '''
```

```
    number_items_per_group = len(list(genes_per_group.values())[0])
    number_groups = len(genes_per_group)
    gene_list = list([item for sublist in list(
        genes_per_group.values()) for item in sublist])

    outdir = os.path.dirname(output_file)
    if not os.path.exists(outdir):
        os.makedirs(outdir, exist_ok=True)

    f, ax = plt.subplots(number_groups, ncols=number_items_per_group,
↪   sharex='col', sharey=False,
                         figsize=(5 * number_items_per_group, 3 *
↪   number_groups))
    f.subplots_adjust(wspace=0.5, hspace=0.1)
    for n, i in enumerate(gene_list):
        ax_n = int(n / number_items_per_group), n - \
            number_items_per_group * int(n / number_items_per_group)
        gene_name = i.replace('_', '-')
        fig = whiskerplot_gene(df, 'group', group_order=group_order, gene=i,
↪   style="boxplot", ax=ax[ax_n],
                               ylabel=gene_name)
        if n == len(gene_list):
            fig.spines['bottom'].set_visible(True)
        else:
            fig.spines['bottom'].set_visible(False)
    f.savefig(output_file, bbox_inches='tight')


def multi_gene_graph(df,
                     genes,
                     group_order,
                     group="group",
                     output_file=None,
                     graph_type="whiskerplot",
                     rows=None,
                     cols=None,
                     ylim=None):
    '''
    Function to create a graph with a defined set of genes

    Parameters:
        df: dataframe with FPKM values
        genes_per_group: dictionary of genes for each group
        group_order: order of groups for subgraph
        output_file: path of output file
```

```
    '''
    if rows == None and cols == None:
        cols = int(math.sqrt(len(genes)))
        rows = math.ceil(len(genes)/cols)

    f, ax = plt.subplots(nrows=rows, ncols=cols, sharex='col', sharey=False,
                         figsize=(5 * cols, 3 * rows))
    f.subplots_adjust(wspace=0.3, hspace=0.1)
    for n, i in enumerate(genes):
        row_n = int(n/cols)
        col_n = n % cols
        fig = whiskerplot_gene(df,
                               group,
                               group_order=group_order,
                               gene=i,
                               ax=ax[row_n, col_n],
                               ylabel=i,
                               ylim=ylim)
        if n == len(genes):
            fig.spines['bottom'].set_visible(True)
        else:
            fig.spines['bottom'].set_visible(False)

    if output_file:
        outdir = os.path.dirname(output_file)
        if not os.path.exists(outdir):
            os.makedirs(outdir, exist_ok=True)
        f.savefig(output_file, bbox_inches='tight')
    plt.show()
    plt.close()


def correlation_gene(df, gene_of_interest, output_file):
    '''
    Function to create a table with the correlation of the gene of
 ↪  interest with the rest of genes

    Parameters:
        df: dataframe with FPKM values
        gene_of_interest: gene of interest
        output_file: path of output file
    '''
    print('Performing correlation of gene ', gene_of_interest)
    gene_of_interest = df[gene_of_interest]
    correlation_list = []
```

```
    for gene in df:
        correlation_list.append(
            [gene] + list(stats.spearmanr(gene_of_interest, df[gene])))
    pd_correlation_list = pd.DataFrame(correlation_list, columns=[
                                       'gene', 'spearman', 'pval'])
    pd_correlation_list['abs_spearman'] =
↪   abs(pd_correlation_list['spearman'])
    pd_correlation_list = pd_correlation_list[pd_correlation_list['pval'] <
↪   0.05]
    pd_correlation_list = pd_correlation_list.sort_values(
        by='abs_spearman', ascending=False).iloc[1:]
    pd_correlation_list.to_csv(output_file, index=False)
    return pd_correlation_list


def gen_groups(df, sample_var='sample', position_group=0):
    df['antibody'] = [i.split("_")[position_group] for i in df[sample_var]]
    df['group'] = df['antibody']
    df['group'] = df['group'].replace(
        to_replace=["hmgcr", "srp"], value="imnm")
    df['group'] = df['group'].replace(
        to_replace=["mi2", "nxp2", "tif1", "mda5"], value="dm")
    df['group'] = df['group'].replace(to_replace=["jo1"], value="as")
    return df


def setup_gene_tmm(tmm_path='./data/salmon/gene_tmm.csv',
↪   position_group=0):
    df = pd.read_csv(tmm_path).set_index('gene_id').T
    df['sample'] = df.index
    df.reset_index(drop=True, inplace=True)
    df = df.rename_axis(None, axis=1)
    df = gen_groups(df, position_group=position_group)
    return df
```

1.4. Visualizador

rnaseq_viewer.R

```
#!/usr/bin/env Rscript

# Viewer for RNAseq data
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022
```

```
pacman::p_load(shiny)
source('./src/server.R')
source('./src/ui.R')

shinyApp(
  ui = ui,
  server = server
)
```

server.R

```
#!/usr/bin/env Rscript

# Viewer for RNAseq data
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022

pacman::p_load(plyr, ggbeeswarm, psych, ggplot2, data.table, stringr,
 ↪  matrixStats, plotly)

refresh_data <- function() {
  list_files <- list.files("./data/", pattern = "*.csv", full.names = T)
  dataset_names <- gsub("\\.csv$","", basename(list_files))
  sort_lists <- strsplit(readLines("./data/sort_order.txt", warn=FALSE),
 ↪  ",")
  datasets <- lapply(list_files, function(file) {
      print(paste("Processing", basename(file)))
      df <- fread(file)
      df <- df[, unique(colnames(df)), with=FALSE]
      var_names <- colnames(df)
      genes <- var_names[unlist(lapply(var_names,
 ↪  str_detect,"[[:upper:]]"))]
      vars <- var_names[unlist(lapply(var_names, str_detect,"[[:lower:]]"))]

      #Sort levels of variables
      for (var in vars) {
        cat(paste(var, " "))
        sorted_var <- sort(unique(df[[var]]))
        for (sort_list in sort_lists) {
          sorted_list <- sort(sort_list)
          if(identical(sorted_var, sorted_list)) {
            df[, (var) := factor(get(var), levels = sort_list)]
          }
        }
      }
```

```
      cat("\n")

      data <- list(df=df, genes=genes, vars=vars)
      class(data) <- "data"
      return(data)
  })

  names(datasets) <- dataset_names

  save(dataset_names, datasets, file = "./data/df.RData")
}

server<-function(input, output, session){

  if(file.exists("./data/df.RData")) {
    load("./data/df.RData")
  } else {
    refresh_data()
    load("./data/df.RData")
  }

  updateSelectInput(session, "dataset", choices=dataset_names, selected =
   ↪  "muscle_biopsies")
  updateSelectizeInput(session, "gene", choices=datasets[[1]]$genes, server
   ↪  = TRUE, selected = "ISG15")
  updateSelectizeInput(session, "gene2", choices=datasets[[1]]$genes, server
   ↪  = TRUE, selected = "ISG15")

  dataset <- reactive({
    datasets[[input$dataset]]
  })

  observeEvent(input$dataset, {
    updateSelectizeInput(session, "group",
     ↪  choices=datasets[[input$dataset]]$vars, selected = "group")
    updateSelectizeInput(session, "hue", choices=c('None',
     ↪  datasets[[input$dataset]]$vars), selected = 'None')
 })

  observeEvent(input$refresh, {
   refresh_data()
  })

  output$nogene <- renderText({
    if(input$gene == ""){
```

```
    "Please, select a valid gene"
  }
})

output$myplot <- renderPlot({
  if(input$gene != ""){
    df <- dataset()$df

    gene <- input$gene
    df[, gene] <- log2(df[, ..gene] + 1)

    if (input$hue == "None") color <- NULL else color <- input$hue

    ggplot(df, aes_string(x=as.name(input$group), y=as.name(input$gene),
     ↪  label=as.name(input$gene), color=color)) +
      geom_boxplot(fill='grey') +
      geom_beeswarm(corral = "gutter", dodge.width = 0.75) +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
            text = element_text(size = 20))
  }
})

output$downplot <- downloadHandler(
  filename = function() {paste(gsub(", ", "_", toString(input$gene)),
   ↪  "_by_", input$group, '.pdf', sep='')},
  content = function(file) {
    if(input$gene != ""){
      df <- dataset()$df

      gene <- input$gene
      df[, gene] <- log2(df[, ..gene] + 1)

      if (input$hue == "None") color <- NULL else color <- input$hue

      ggplot(df, aes_string(x=as.name(input$group), y=as.name(input$gene),
       ↪  label=as.name(input$gene), color=color)) +
        geom_boxplot(fill='grey') +
        geom_beeswarm(corral = "gutter", dodge.width = 0.75) +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust =
         ↪  1),
              text = element_text(size = 20))
      ggsave(file)
    }
```

```
  }
)

output$noraw <- renderText({
  if(length(input$gene) < 1){
    "No gene to show results"
  }})

output$raw_data <- renderDataTable({
  if(input$gene != ""){
    df <- dataset()$df
    gene <- input$gene
    df[, gene] <- log2(df[, ..gene] + 1)
    df <- df[, c("sample", gene), with=FALSE]
    colnames(df) <- c("id", paste0("log2TMM(", gene, "+1)"))
    df <- as.data.frame(df)
    df
  }})

output$downraw<-downloadHandler(
  filename = function() { paste(gsub(", ", "_", toString(input$gene)),
   ↪  "_by_", input$group, '.csv', sep='') },
  content=function(file){
    df <- df()
    var_list <- df[,c(unlist(strsplit(input$gene, split=" ")), "DX")]
    write.csv(var_list, file)
  }
)

output$nogene2 <- renderText({
  if(input$gene == "" | input$gene2 == ""){
    "Please, select a valid gene"
  }
})

output$my2gene <- renderPlotly({
  if(input$gene != "" & input$gene2 != ""){
    df <- dataset()$df

    gene <- input$gene
    gene2 <- input$gene2
    df[, gene] <- log2(df[, ..gene] + 1)
    df[, gene2] <- log2(df[, ..gene2] + 1)

    if (input$hue == "None") color <- NULL else color <- input$hue
```

```
      p <- ggplot(df, aes_string(x=as.name(input$gene),
↪  y=as.name(input$gene2), color=color, text='sample')) +
        geom_point() +
        theme_minimal()
      ggplotly(p, tooltip="text")
    }
  })

  output$gene_corr <- renderDataTable({
    if(input$gene != ""){
      df <- dataset()$df
      gene <- input$gene
      df_cor <- cor(df[, ..gene], df[, dataset()$genes, with=FALSE],
↪  method="spearman")[1,]
      df_cor <- sort(df_cor, decreasing = TRUE)
      data.frame(cbind(names = names(df_cor), genes=df_cor))
    }})

  output$myqc <- renderPlotly({
    datasets <- dataset()
    df <- datasets$df[, datasets$genes, with=FALSE]
    myvars <- apply(df,2, var,na.rm=TRUE)
    myvars <- sort(myvars,decreasing=TRUE)
    myvars <- myvars[1:500]
    myvars <- names(myvars)
    df <- data.frame(df[, ..myvars])
    df_high_var <- log2(df + 1)
    pca <- prcomp(df_high_var, center=TRUE, scale=FALSE, rank. = 2)
    p <- ggplot(data.frame(pca$x), aes(x=PC1, y=PC2,
↪  text=datasets$df[['sample']])) +
      geom_point(aes(color=datasets$df[[input$group]])) +
      labs(color=input$group) +
      theme_minimal()

    ggplotly(p, tooltip="text")
  })

  output$gene_info <- renderUI({
    url <-
↪  paste0("https://www.genecards.org/cgi-bin/carddisp.pl?gene=",input$gene)
    tags$iframe(src = url, style='width:66vw;height:100vh;')
  })
}
```

ui.R

```
#!/usr/bin/env Rscript

# Viewer for RNAseq data
# Author: Iago Pinal-Fernandez
# Copyright (c) Iago Pinal-Fernandez, 2022

ui<-fluidPage(
  titlePanel(title="RNAseq muscle biopsies"),
  sidebarLayout(
    sidebarPanel(
      selectInput("dataset", "Select dataset of interest", choices = NULL),
      selectizeInput("group", "Select group of interest", choices = NULL),
      selectizeInput("gene", "Select gene of interest", choices = NULL),
      selectizeInput("hue", "Select hue", choices = NULL),
      selectizeInput("gene2", "Select gene 2", choices = NULL),
      br(),
      h4("Muscle Disease Group"),
      h4("Data Viewer"),
      p(em("Iago Pinal-Fernandez 2022")),
      actionButton("refresh", "Refresh")
    ),
    mainPanel(
      tabsetPanel(type="tab",
        tabPanel("Plot",
         textOutput(outputId="nogene"),
         plotOutput(outputId="myplot"),
         downloadButton(outputId="downplot",
                        label="Download the plot")
        ),
        tabPanel("Raw Data",
         textOutput(outputId="noraw"),
         dataTableOutput("raw_data"),
         downloadButton(outputId="downraw",
                         label="Download the raw data")
        ),
        tabPanel("Two-gene comparisons",
                 textOutput(outputId="nogene2"),
                 plotlyOutput(outputId="my2gene")
        ),
        tabPanel("Gene correlation",
                 dataTableOutput("gene_corr")
        ),
        tabPanel("Quality control",
```

```
                plotlyOutput(outputId="myqc")
        ),
        tabPanel("Gene info",
                uiOutput("gene_info")
        )
      )
    )
  )
)
```

2. Artículos generados

2.1. Miopatía por inhibidores de *checkpoint*

- Los inhibidores de *checkpoint* pueden causar miositis, a menudo acompañada de miastenia grave o de un fenotipo similar a la miastenia grave y/o miocarditis.
- Nuestros análisis transcriptómicos identificaron tres tipos distintos de ICI-miositis (ICI-DM, ICI-MYO1 e ICI-MYO2).
- La vía de IL6 se sobreexpresó en todos los grupos y la vía de interferón tipo 2 se activó tanto en ICI-DM como en ICI-MYO1.
- ICI-DM incluyó pacientes con DM y autoanticuerpos anti-TIF1g que, al igual que los pacientes con DM, sobreexpresaron genes inducibles por interferón tipo 1.
- Los pacientes con ICI-MYO1 tenían biopsias musculares altamente inflamatorias e incluyeron a todos los pacientes que desarrollaron miocarditis.
- ICI-MYO2 está compuesta por pacientes con miopatía necrosante y bajos niveles de inflamación muscular.
- La ICI-miositis está compuesta de tres grupos clínicos de pacientes, con diferentes características transcriptómicas y probablemente diferente patogenia.
- Hemos identificado vías terapéuticas relevantes que pueden útiles para tratar este evento adverso autoinmune.

Transcriptomic profiling reveals distinct subsets of immune checkpoint inhibitor-induced myositis

Iago Pinal-Fernandez[1,2*], Angela Quintana[1,3*], Jose C. Milisenda[1,4,5*], Maria Casal-Dominguez[1,2*], Sandra Muñoz-Braceras[1], Assia Derfoul[1], Jiram Torres-Ruiz[1], Katherine Pak[1], Stefania Del Orso[1], Faiza Naz[1], Gustavo Gutierrez-Cruz[1], Margherita Milone[6], Shahar Shelly[7], Yaiza Duque-Jaimez[4], Ester Tobias-Baraja[4], Ana Matas-Garcia[4,5], Gloria Garrabou[4,5], Joan Padrosa[5], Javier Ros[8], Ernesto Trallero-Araguás[9], Brian Walitt[10], Lisa Christopher-Stine[2,11], Thomas E. Lloyd[2], Chen Zhao[12], Shannon Swift[12], Arun Rajan[12], Josep Maria Grau[4,5], Albert Selva-O'Callaghan[3,13], Teerin Liewluck[6#], Andrew L. Mammen[1,2,11#]

1- Muscle Disease Unit, National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, Bethesda, MD, USA
2- Department of Neurology, Johns Hopkins University School of Medicine, Baltimore, MD, USA
3- Systemic Autoimmune Disease Unit, Vall d'Hebrón Institute of Research, Barcelona, Spain
4- Muscle Research Unit, Internal Medicine Service, Hospital Clinic, Barcelona, Spain
5- CIBERER, Barcelona, Spain
6- Division of Neuromuscular Medicine, Department of Neurology, Mayo Clinic, Rochester, MN, USA

7- Department of Neurology, Rambam Health Care Campus, Bruce Rappaport Faculty of Medicine, Technion-Israel Institute of Technology, Haifa, Israel.
8- Medical Oncology, Vall d'Hebrón Hospital, Barcelona, Spain
9- Rheumatology Department, Vall d'Hebron Hospital, Barcelona, Spain.
10- Division of Intramural Research, Department of Health and Human Services, National Institute of Nursing Research, National Institutes of Health, Bethesda, MD, USA.
11- Department of Medicine, Division of Rheumatology, Johns Hopkins University School of Medicine, Baltimore, MD, USA
12- Thoracic and Gastrointestinal Malignancies Branch, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda, MD, USA
13- Universitat Autónoma de Barcelona, Barcelona, Spain.

*,# These authors contributed equally to this project.

Address correspondence to: Andrew L. Mammen, M.D., Ph.D., or Iago Pinal-Fernandez, M.D., Ph.D. Muscle Disease Unit, Laboratory of Muscle Stem Cells and Gene Regulation, National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, 50 South Drive, Room 1141, Building 50, MSC 8024, Bethesda, MD 20892. E-mail: andrew.mammen@nih.gov or iago.pinalfernandez@nih.gov. Phone: 301-451-1199. Fax: 301-594-0305.

Competing interests: None

Contributorship: All authors contributed to the development of the manuscript, including interpretation of results, substantive review of drafts and approval of the final draft for submission.

Acknowledgments: We would like to acknowledge Helena Verdaguer for her kind contribution to this research.

Funding: This study was funded, in part, by the Intramural Research Programs of the National Institute of Arthritis and Musculoskeletal and Skin Diseases (NIAMS), the National Cancer Institute (NCI), the Center for Cancer Research of the National Institutes of Health, and through a Cooperative Research and Development Agreement between the NCI and EMD Serono Research & Development Institute, Inc., Billerica, MA, USA, an affiliate of Merck KGaA (CrossRef Funder ID:10.13039/100004755), as part of an alliance between Merck and Pfizer. This work was also supported by the Peter Buck and the Huayi and Siuling Zhang Discovery Fund.

Ethical approval information: All biopsies were from subjects enrolled in institutional review board (IRB)-approved longitudinal cohorts in the National Institutes of Health, the Mayo Clinic, the Clinic Hospital, or the Vall d'Hebron Hospital.

Data sharing statement: Any anonymized data not published within the article will be shared by request from any qualified investigator.

Patient and public involvement: Patients and/or the public were not involved in the design, or conduct, or reporting, or dissemination plans of this research.

Keywords

Immune checkpoint inhibitors, myositis, immune-related adverse events, anti-PD-1, anti-PD-L1, IL6, IFNG, interferon

KEY MESSAGES

What is already known about this subject?

- Immune checkpoint inhibitors can cause myositis which is often accompanied by myasthenia gravis or a myasthenia gravis-like phenotype, and/or myocarditis.

What does this study add?

- Transcriptomic analyses identified three distinct types of ICI-myositis (ICI-DM, ICI-MYO1, and ICI-MYO2).
- The IL6 pathway was overexpressed in all groups, and the type 2 interferon pathway was activated both in ICI-DM and ICI-MYO1.
- ICI-DM included patients with DM and anti-TIF1γ autoantibodies who, like DM patients, overexpressed type 1 interferon-inducible genes.
- ICI-MYO1 patients had highly inflammatory muscle biopsies and included all patients that developed co-existing myocarditis.
- ICI-MYO2 was composed of patients with predominant necrotizing pathology and low levels of muscle inflammation.

How might this impact on clinical practice?

- ICI-myositis is composed of different clinical groups of patients, with different transcriptomic features and likely different pathogenesis.
- We have identified relevant therapeutic pathways that may be targeted to treat this autoimmune adverse event.

Abstract

Objectives: Inflammatory myopathy or myositis is a heterogeneous family of immune-mediated diseases including dermatomyositis (DM), antisynthetase syndrome (AS), immune-mediated necrotizing myopathy (IMNM), and inclusion body myositis (IBM). Immune checkpoint inhibitors (ICI) can also cause myositis (ICI-myositis). This study was designed to define gene expression patterns in muscle biopsies from patients with ICI-myositis.

Methods: Bulk RNA sequencing was performed on 200 muscle biopsies (35 ICI-myositis, 44 DM, 18 AS, 54 IMNM, 16 IBM, and 33 normal muscle biopsies) and single nuclei RNA sequencing was performed on 22 muscle biopsies (7 ICI-myositis, 4 DM, 3 AS, 6 IMNM, and 2 IBM).

Results: Unsupervised clustering defined three distinct transcriptomic subsets of ICI-myositis: ICI-DM, ICI-MYO1, and ICI-MYO2. ICI-DM included patients with DM and anti-TIF1γ autoantibodies who, like DM patients, overexpressed type 1 interferon-inducible genes. ICI-MYO1 patients had highly inflammatory muscle biopsies and included all patients that developed co-existing myocarditis. ICI-MYO2 was composed of patients with predominant necrotizing pathology and low levels of muscle inflammation. The type 2 interferon pathway was activated both in ICI-DM and ICI-MYO1. Unlike the other types of myositis, all three subsets of ICI-myositis patients overexpressed genes involved in the IL6 pathway.

Conclusions: We identified three distinct types of ICI-myositis based on transcriptomic analyses. The IL6 pathway was overexpressed in all groups, the type I interferon pathway activation was specific for ICI-DM, the type 2 IFN pathway was overexpressed in both ICI-DM and ICI-MYO1, and only ICI-MYO1 patients developed myocarditis.

Introduction

Inflammatory myopathy or myositis is a heterogeneous family of immune-mediated diseases that includes dermatomyositis (DM), antisynthetase syndrome (AS), immune-mediated necrotizing myopathy (IMNM), and inclusion body myositis (IBM).[1] Each of these is associated with distinctive clinical features and muscle biopsies from each type of myositis have unique histopathological features and gene expression profiles.[2]

In recent years, it has been recognized that immune checkpoint inhibitors (ICI) can trigger a novel form of myositis in cancer patients (ICI-myositis). Muscle biopsies from ICI-myositis patients are characterized by the presence of T cells and macrophages with minimal B cell infiltration.[3-6] Unlike patients with DM, AS, IMNM, or IBM, ICI-myositis is often accompanied by myasthenia gravis or a myasthenia gravis-like phenotype,[7] and/or myocarditis. Furthermore, a minority of patients with ICI myositis also have a DM-like skin rash. However, it remains unknown whether muscle biopsies from ICI myositis patients have a unique transcriptomic profile and whether biopsies from those patients with and without DM-like rashes have different gene expression patterns. In this study, we addressed these questions by performing bulk and single-nuclei RNA sequencing on muscle biopsy tissue obtained from patients with ICI-myositis and comparing them to other types of myositis.

Methods

Patients and samples

Thirty-five patients from the National Institutes of Health (n=4, Bethesda, US), Mayo Clinic (n=11, Rochester, US), Vall d'Hebron Hospital (Barcelona, Spain), and Clinic Hospital (n=20, Barcelona, Spain) with a myopathy related to ICI treatment (either anti-PD-1 or anti-PD-L1 therapy alone or in combination with anti-CTLA-4) and an available frozen muscle biopsy were included in the study. ICI-myositis was defined as the presence of muscle weakness, or the presence of myopathic changes with or without prominent inflammatory infiltrate in the muscle biopsy after starting treatment with ICI.

The epidemiologic features of the patients, type and cycle of ICI, clinical features (including the presence or absence of myocarditis, dysphagia, and ocular involvement), autoantibody profile (including myositis-specific [anti-HMGCR, anti-SRP, anti-Mi2, anti-NXP2, anti-TIF1g, anti-MDA5, anti-Jo1], myositis-associated autoantibodies [anti-Ro52, anti-NT5c1a, anti-PM/Scl], anti-AChR, anti-Musk, and anti-striational autoantibodies) were included in the analysis.

The biopsies from patients with ICI-myositis were compared to 33 normal muscle biopsies and 132 muscle biopsies from patients with the most common types of inflammatory myopathy, including 54 IMNM, 44 DM, 18 AS, and 16 IBM. Of note, the transcriptomic profiles of the DM, AS, IMNM, and IBM muscle biopsies have been previously reported.[2,

8-11] All biopsies were from subjects enrolled in institutional review board (IRB)-approved longitudinal cohorts in the different hospitals.

The transcriptional characteristics of the muscle biopsies were also compared with publicly available transcriptomic datasets of tumors before and after ICI treatment (GEO: GSE91061).[12]

Differentiating human skeletal muscle myoblasts treated with different types of interferon

We treated human skeletal muscle myoblasts (HSMMs) to identify the IFNB1 and IFNG-specific interferon-stimulated genes. Normal HSMMs were cultured according to recommended protocol by the manufacturer (Lonza). When 80% confluent, the cultures were induced to differentiate into myotubes by replacing the growth medium with differentiation medium (Dulbecco's modified Eagle's medium supplemented with 2% horse serum and L-glutamine). Differentiating HSMMs were treated daily with 100U/L and 1000U/L of IFNA2a (R&D, ref:11100-1), IFNB1 (PeproTech, ref:300-02BC), or IFNG (PeproTech, ref:300-02) for 7 days and then harvested for RNA extraction and subsequent RNA sequencing.

Bulk RNA sequencing

Bulk RNA sequencing was performed on frozen muscle biopsy specimens as previously described.[2, 8-10] In short, RNA was extracted from fresh-frozen biopsies using TRIzol (Invitrogen) and quantified using NanoDrop. Libraries for bulk RNA sequencing were prepared using NEBNext Poly(A) mRNA Magnetic Isolation Module and Ultra™ II Directional

RNA Library Prep Kit for Illumina (New England BioLabs, cat. #E7490 and #E7760). The input RNA and the resulting libraries were analyzed with Agilent 4200 Tapestation for quality assessment. The libraries were sequenced using the NextSeq 550 and the NovaSeq 6000 Illumina platforms.

Single-nuclei RNA-sequencing

We performed single-nuclei RNA sequencing in 4 ICI-myositis, 3 ICI-dermatomyositis (ICI-DM), 4 DM, 3 AS, 6 IMNM, and 2 IBM. For the nuclei isolation, we used a modification of the sucrose-gradient ultracentrifugation nuclei isolation protocol from Schirmer et al.[13] Ten mg of frozen muscle tissue was sectioned and homogenized in 1mL of lysis buffer (0.32M sucrose, 5mM CaCl2, 3mM MgCl2, 0.1mM EDTA, 10mM Tris-HCl pH 8, 1mM DTT, 0.5% Triton X-100 in DEPC-treated water) using 1.4mm ceramic beads low-binding tubes and the Bertin Technology Precellys 24 lysis homogenizer (6500rpm-3times x 30s). The homogenized tissue was transferred into open-top thick-walled polycarbonate ultracentrifuge tubes (25x89 mm, Beckman Coulter) on ice. 3.7mL of sucrose solution (1.8 M sucrose, 3mM MgCl2, 1mM DTT, 10mM Tris-HCl) were pipetted to the bottom of the tube containing lysis buffer generating two separated phases (sucrose on the bottom and homogenate on the top). The tubes were filled almost completely with lysis buffer and weighted for balance. The samples were ultracentrifuged (Beckman Coulter XE-90, SW32 rotor, swinging bucket) at 24,400rpm (107,163rcf) for 2.5 hours at 4ºC, transferred to ice, and the supernatant removed. Two hundred microliters of DEPC-PBS were added to each pellet, incubated on ice for 20 minutes, and then pellets were resuspended. The resulting samples were filtered twice using 30μm Miltenyi pre-separation filters. The nuclei were

counted using a manual hemocytometer. Between 2000 and 3000 nuclei per sample were loaded in the 10X Genomic Single-Cell 3' system. We performed the 10X nuclei capture and the library preparation protocol according to the manufacturer's instructions without modification.

Statistical analysis

Bulk RNAseq reads were demultiplexed using bcl2fastq/2.20.0 and preprocessed using fastp/0.21.0. The abundance of each gene was generated using Salmon/1.5.2 and quality control output was summarized using multiqc/1.11. Dimensionality reduction was performed with the uniform manifold approximation and projection (UMAP) using umap/0.2.9.0. The number of clusters was determined with Tibshirani's gap statistic[14] using factoextra/1.0.7. The clusters were defined using the K-means algorithm. Counts were normalized using the Trimmed Means of M values (TMM) from edgeR/3.34.1 for graphical analysis. Differential expression was performed using limma/3.48.3. Pathway analysis was done using Gene Set Enrichment Analysis (GSEA) using clusterProfiler/4.6.0 and GSEA/4.2.3 for the Reactome and the Hallmark datasets.[15]

For the single-cell and single-nuclei RNAseq, reads were demultiplexed and aligned using cellranger/6.0.1. The samples were aggregated, normalized (SCTransform), and integrated (RunHarmony) using Seurat/4.1.0. Graphical analysis of single cell and single nuclei RNAseq data used the functions contained in Seurat/4.1.0.

The Benjamini-Hochberg correction was used to adjust for multiple comparisons and a corrected p-value (q-value) of 0.05 or less was considered statistically significant. Graphical analysis used both the Python and R programming languages.

Results

Clinical features of patients with ICI-myositis

Thirty-five patients (11 female, average age 67yo) with ICI-myositis were included in the study and three (8.6%) of these had DM rashes. Twenty-four (69%) were treated with PD1 inhibitors, and 11 (31%) with PD-L1 inhibitors. In addition, five (14%) were concomitantly treated with CTLA-4 inhibitors. The three most prevalent primary tumors were melanoma (n=10), thymoma (n=5), and lung cancer (n=4). Sixteen (46%) developed myopathy after receiving their first cycle of ICI, and the rest of them after two or more cycles. Seven ICI-myopathy patients (20%) had diplopia and 13 (37%) of them developed myocarditis. Twenty-nine (83%) patients had autoantibodies against the neuromuscular junction or skeletal muscle antigens: 14 had anti-acetylcholine receptor (AChR) antibodies, 15 had anti-striational antibodies, and 16 were positive for various other myositis autoantibodies. The three patients with DM rashes had high titers of anti-TIF1γ autoantibodies. In three patients with pre-ICI treatment serum available, the same autoantibodies that were present after treatment with ICI were detectable before ICI initiation (two patients with anti-AChR and anti-striational autoantibodies and one patient with DM rashes and anti-TIF1γ autoantibodies). The complete characteristics of these patients and their clinical evolution are described in Table 1 and Supplemental Table S1. The clinical features of the DM, AS, IMNM, and IBM subjects have been previously reported.[2, 8-11]

Unsupervised clustering reveals three transcriptomically distinct groups of ICI-myositis patients

To determine whether distinct subtypes of ICI-myositis could be defined based on transcriptomic data from muscle biopsies, we performed unsupervised clustering analysis using the expression levels of genes in the different muscle biopsies. This revealed three distinct clusters of biopsies and patients in each cluster had unique clinical features as well. The ICI-DM cluster (n=3) included the three patients with DM rashes ($p<0.001$). The ICI-MYO1 cluster (n=24) included all the patients who developed myocarditis and patients in this cluster had a higher prevalence of anti-AChR autoantibodies (50% vs. 25%) compared to patients in the ICI-MYO2 cluster (n=8) (Figure 1, Table 1, Supplementary Table 1).

Muscle biopsies from two patients in the ICI-DM cluster had perifascicular atrophy and the third had myofiber necrosis with intense perivascular inflammation. Muscle biopsies from patients in the ICI-MYO1 cluster were highly inflammatory compared to those in the ICI-MYO2 cluster, where necrosis was the predominant muscle biopsy feature. The inflammatory infiltrates of biopsies from the ICI-MYO1 cluster were predominantly composed of macrophages and CD8+ cells (Figure 2).

All the single-nuclei RNAseq analyses were performed using muscle biopsies from patients in the ICI-MYO1 cluster (Supplementary Figure 1).

Type 2 interferon-inducible genes are overexpressed in ICI-MYO1 and ICI-DM

IFNγ and IFNγ-inducible genes were robustly overexpressed in patients with both ICI-MYO1 (e.g. GBP2 log2 fold-change[FC] 2.5, q-value 1.2e-13) and ICI-DM (e.g. GBP2 log2

fold-change[FC] 3, q-value 1.2e-11). The expression levels of these genes were comparable to that observed in patients with AS, and IBM. Biopsies in the ICI-MYO2 cluster were significantly lower than in ICI-DM or ICI-MYO1, but still had significantly higher levels of these genes compared to normal biopsies (GBP2 log2 fold-change[FC] 1.2, q-value 6e-6) (Figure 3-4, Supplementary Figure 2-4, Supplementary Table 3).

Although single-nuclei RNA-seq identified minimal levels of IFNG in T-cells, genes specifically upregulated by IFNG were, in general, below the detection threshold of this technique (Figure 5, Supplementary Table 4-5).

IL6 pathway genes are specifically up-regulated in all patients with ICI-myositis. Pathway analysis showed overexpression of signaling by interleukins (Figure 4). Of all the interleukins, genes associated with the IL6 pathway were the most specifically overexpressed in muscle biopsies from patients with ICI-MYO1, ICI-MYO2, and ICI-DM. These included genes encoding the IL6 receptor (IL6R), STAT3, and TYK2 (Figure 3-4, Supplementary Figure 5-7, Supplementary Table 3). Furthermore, IL6 expression itself was positively correlated with the level of expression of canonical inflammatory T-cell markers including CD4 and CD8 as well as the macrophage markers CD14 and CD68 (Supplementary Figure 8).

The gene encoding CEBPB, which binds to regulatory regions of several acute-phase and cytokines genes, including IL6, was also specifically overexpressed in all patients with ICI-myositis (ICI-MYO1, ICI-MYO2, and ICI-DM). Also, genes implicated in the suppression of

the IL6 pathway, like SOCS3,[16] were more activated in ICI-MYO1 than in ICI-MYO2 (Figure 3, Supplementary Figure 6, Supplementary Table 3).

The expression levels of JUN, FOS, and EGR1 correlated with the expression of IL6 (Supplementary Figure 9) and were overexpressed in all three ICI-myositis clusters (ICI-MYO1, ICI-MYO2, and ICI-DM) (Supplementary Figure 10). In general, the overexpression of these genes was more intense in ICI-MYO1 and ICI-DM than in ICI-MYO2 (Supplementary Figure 10). Both EGR1, and the members of the transcription factor complex Activator Protein-1 JUN and FOS, are regulators of the IL6 pathway.[17-20]

Single-nuclei data confirmed that the IL6 pathway and its regulators were robustly overexpressed both in ICI-MYO1 and ICI-DM (Figure 6, Supplementary Tables 4-5). This study also showed that IL6R was expressed primarily in macrophages (Figure 6), although its expression was at levels too low to detect differences between groups (Supplementary Tables 4-5).

Type 1 interferon-inducible genes are overexpressed in ICI-DM

Compared to patients in the ICI-MYO1 or ICI-MYO2 clusters, the three patients within the ICI-DM cluster had a marked elevation of type I interferon-stimulated genes, as seen in DM patients (Figure 3-4, Supplementary Figure 11-12). In these ICI-DM patients, IFNB1 gene expression was detectable (Supplementary Figure 13). In contrast, other type-I interferon genes were either undetectable or present at lower levels, as in muscle biopsies from patients with DM (Supplementary Figure 13, Supplementary Table 3).

Single-nuclei RNAseq verified a robust activation of the type 1 interferon pathway affecting all the different cell types of patients with ICI-DM (Figure 5, Supplementary Tables 4-5)

Other transcriptomic features of ICI-myositis compared to DM, AS, IMNM, IBM, and healthy comparator muscle.

Compared to control muscle tissue, muscle biopsies from patients with ICI-MYO1 and ICI-DM had increased expression of genes associated with macrophages (CD14, and CD68) and T-cells (CD3E, CD4, CD8A, PRF1, GZMA, GZMB) along with reductions in the expression of skeletal muscle structural genes (MYH1, ACTA1). Genes upregulated during muscle regeneration (NCAM1, MYH3) were elevated in both ICI-MYO1 and ICI-DM. Genes associated with oxidative phosphorylation and mitochondrial genes had decreased expression; this was more pronounced in ICI-MYO1 and ICI-DM than in ICI-MYO2. The expression of immunoglobulin genes was increased only in ICI-MYO1; levels of these genes were similar to those in DM and AS, but lower than in IBM (Supplementary Figure 14-17, Supplementary Table 3).

Multiple TNF receptors and their ligands were exclusively overexpressed in patients with ICI, including those signaling TNFa (Supplementary Figure 18-20, Supplementary Table 3). Accordingly, the TNFa pathway was overexpressed in patients with ICI-MYO1 and ICI-DM, similar to other types of inflammatory myopathy.

Also, vascular adhesion molecules like VCAM1 and ICAM1 were overexpressed at similar levels to other types of inflammatory myopathy in ICI-DM and ICI-MYO1, but not in ICI-MYO2 (Supplementary Figure 21, Supplementary Table 3).

Finally, several checkpoint genes such as PDCD1, and CTLA4 were upregulated in ICI-MYO and ICI-DM, but not in ICI-MYO2 (Supplementary Figure 22).

Comparing transcriptomic profiles of ICI-myositis patients with and without ocular and cardiac involvement

We did not find any significant transcriptomic differences between ICI-myositis patients with and without diplopia or between ICI-myositis patients exposed to different types of ICI (PD-1 inhibitors, PD-L1 inhibitors, and co-treatment with CTLA4). However, patients with myocarditis were all part of cluster ICI-MYO1 and had higher levels of IFNG (log2FC 2.8, q-value 0.03), CD8A (log2FC 2.3, q-value 0.03), CD14 (log2FC 1.9, q-value 0.05), and vascular adhesion molecules (ICAM1 log2FC 1.3, q-value 0.05) (Figure 7, Supplementary Table 3).

Tumors treated with ICI upregulate IFNγ-stimulated genes but not genes of the IL6 pathway

We were interested in whether ICI might cause upregulation of the IL6 pathway even in the absence of myositis. Unfortunately, muscle biopsies from patients who were treated with ICI but who did not develop myositis were not available. However, publicly available RNA sequencing data from melanomas before and after treatment with ICI (GEO: GSE91061) showed overexpression of IFNG (log2FC 1.2, q-value=0.008) and IFNG-stimulated genes after treatment with ICI (e.g. GBP2 log2FC 0.8, q-value=0.03). Unlike the muscle of patients with ICI-myositis, these tumors did not show a significant overactivation of the IL6

pathway after treatment with ICI (e.g. IL6R log2FC 0.008, q-value 0.8, Supplementary Figure 23).

Discussion

In this study, we identified three transcriptomically distinct clusters of ICI-myositis patients and showed that each cluster includes patients with unique clinical features. At the transcriptomic level, all three clusters were characterized by over-expression of the IL6 pathway. In contrast, only biopsies from the ICI-DM and ICI-MYO1 clusters had high expression of IFNγ-stimulated genes, whereas IFNB1 and IFNβ1-inducible genes were only highly expressed in biopsies from the ICI-DM cluster. From the clinical perspective, only patients in the ICI-MYO1 cluster developed myocarditis and only patients in the ICI-DM cluster had DM rashes.

The pathogenesis of autoimmune adverse events in the context of immune checkpoint inhibitor treatment is still not completely understood. One hypothesis is that immune-related adverse events may be caused by a sudden and intense activation of already-existing autoimmunity. In this study, we provide two pieces of information supporting this theory. First, the autoantibodies present at the time of ICI-myositis were also present before ICI treatment in all three patients for whom pre-treatment sera were available. Second, patients with ICI-DM recapitulated the key transcriptomic features of patients with dermatomyositis and had the characteristic autoantibodies of patients with paraneoplastic dermatomyositis (anti-TIF1γ).

Of note, our group has previously shown that in patients with thymoma and ICI-myositis, anti-AChR autoantibodies are detectable prior to the start of ICI therapy.[21] Interestingly, the most common autoantibodies detected in the patients of our study, anti-AChR, and anti-

TIF1g autoantibodies, are also common autoantibodies associated with cancer in patients with myasthenia gravis and myositis,[22, 23] suggesting the possibility that the preexisting autoimmune phenomenon that is activated by ICI may have been directly induced by the tumor itself prior to ICI therapy.

ICI-myositis patients in cluster ICI-MYO1 had frequent myocarditis and autoantibodies targeting the neuromuscular junction (anti-AChR autoantibodies) or skeletal muscle (anti-striated muscle autoantibodies). These features are uncommon in patients with other types of inflammatory myopathy and have also been described in cases of myasthenia gravis not related to ICI.[24] Moreover, many of these patients had clinical features typical of myasthenia gravis, such as diplopia. Autoantibodies in myasthenia gravis bind to the surface of the muscle (postsynaptic receptors of the neuromuscular junction), and thus, these autoantibodies bound to the muscle fibers may have a role attracting to the muscle autoreactive T cells activated by the ICI. Supporting this theory, we have found that patients that develop myocarditis are the ones that have the most active expression of IFNG-inducible genes, T-cell markers, and vascular adhesion molecules.

Our findings may have potential therapeutic implications. Corticosteroids as well as other agents such as IVIG[25] and abatacept[26] have been used to treat ICI-myositis. However, it remains unknown how effective they are. Furthermore, in the case of abatacept, there is reason to be concerned that its binding to CD80/86 may counteract the beneficial effects of ICI therapy on the tumor. In this study, we identified the IL6 pathway as being specifically overexpressed in ICI-myositis and not elevated in tumors treated with ICI. Supporting our

findings, the IL6 pathway has been shown to be elevated in the affected tissue of patients with colitis induced by ICI.[27] Furthermore, tocilizumab, a blocker of the IL6 receptor, has been reported as a potentially effective treatment for various ICI-triggered immune adverse events.[27, 28] Interestingly, it was reported that blocking this pathway may have beneficial effects on the antineoplastic effects of ICI.[27] Thus, it is possible that the overexpression of the IL6 pathway is a general phenomenon not restricted to muscle in ICI autoimmune adverse events, and that treatment with tocilizumab may be useful not only to treat or prevent the adverse event but also to improve tumor prognosis. The fact that type 2 IFN is overexpressed in all ICI-MYO, and type 1 IFN in ICI-DM, also suggests that targeting the JAK-STAT pathway may be useful in patients with ICI-myositis, but risks negatively impact the antineoplastic effect of the drug. Notwithstanding this, given the quick effect of JAK-STAT inhibitors, it may be reasonable to use them in patients with severe cases of ICI-myositis.

This study has several limitations. First, most patients did not have sera available before the start of ICI. Furthermore, the techniques use to assess serologic profiles varied between the different participating centers. Also, there was heterogeneity among patients both in the type of tumors they had and the type of ICI they received. Furthermore, given the severity of their clinical manifestations, half of the patients were treated with corticosteroids before the biopsy was performed (median duration 6 days), which may have reduced the sensitivity of our analyses. These caveats notwithstanding, the lack of heterogeneity in the relatively large number of samples studied may also make our conclusions more generalizable. Finally, compared to bulk RNAseq, single-nuclei RNA sequencing may not detect genes expressed

at low levels and this limited our ability to explore some of the affected pathways in greater detail.

Despite these limitations, this study reveals the existence of three transcriptomically distinct types of ICI-myositis and demonstrates that each type has distinct clinical features. We also demonstrate that the IL6 pathway is activated in all three types of ICI-myositis but not in other types of ICI-naive myositis such as DM, AS, IMNM, or IBM. Based on the evidence provided by this study and previously published studies, we propose that targeting the IL6 pathway may be therapeutically useful in all three types of ICI-myositis patients.

Figure 1. Groups of patients (ICI-MYO1, ICI-MYO2, ICI-DM) resulting from applying unsupervised clustering to the bulk RNA sequencing data of patients with immune checkpoint-induced (ICI) myopathy. Uniform manifold approximation and projection (UMAP) was used to perform the clustering, the number of clusters was determined using the gap statistic and the clusters were defined using the K-means algorithm.

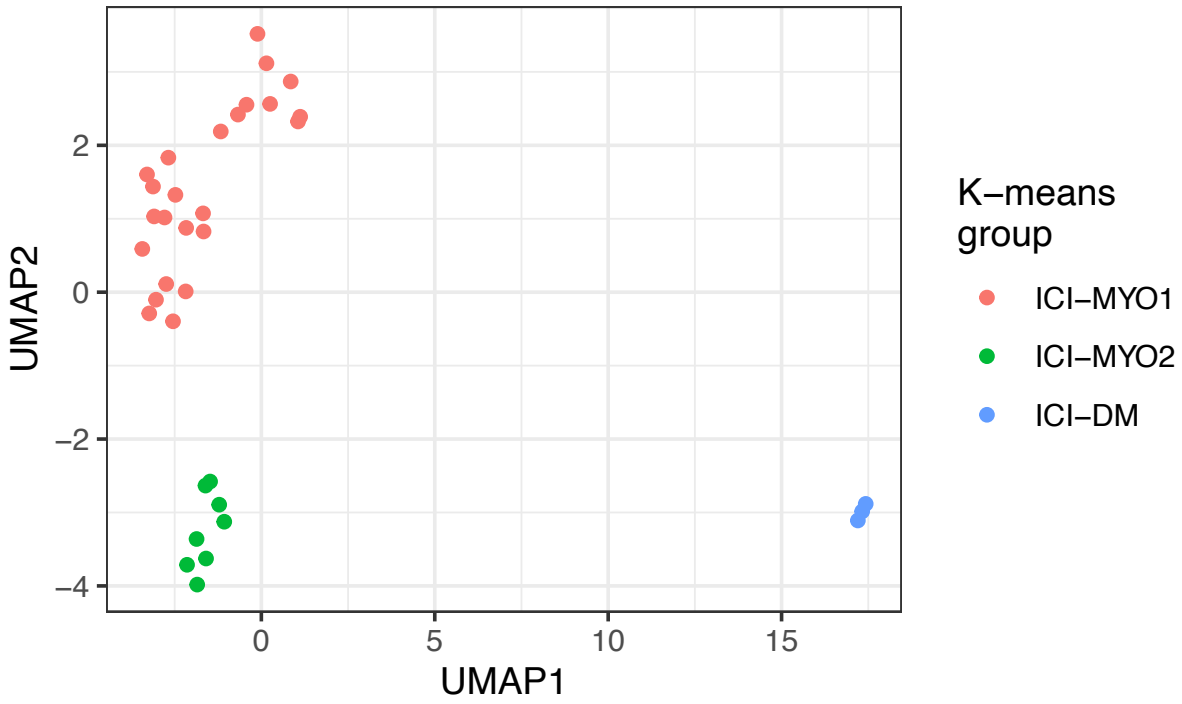


Table 1. General features of immune-checkpoint induced myopathy groups resulting from unsupervised clustering (ICI-DM, ICI-MYO1, ICI-MYO2).

Characteristic	Overall, N = 35[1]	ICI-DM, N = 3[1]	ICI-MYO1, N = 24[1]	ICI-MYO2, N = 8[1]	p-value[2]
Female	11 (31%)	1 (33%)	8 (33%)	2 (25%)	>0.9
Age at biopsy	67 (60, 74)	73 (58, 74)	64 (60, 72)	69 (67, 72)	0.6
Dermatomyositis	3 (8.6%)	3 (100%)	0 (0%)	0 (0%)	**<0.001**
Inflammatory infiltrates	25 (71%)	3 (100%)	19 (79%)	3 (38%)	0.056
Treatment cycle					>0.9
1	16 (46%)	1 (33%)	11 (46%)	4 (50%)	
2	13 (37%)	2 (67%)	8 (33%)	3 (38%)	
3	5 (14%)	0 (0%)	4 (17%)	1 (12%)	
4	1 (2.9%)	0 (0%)	1 (4.2%)	0 (0%)	
PD1 inhibitor	24 (69%)	2 (67%)	14 (58%)	8 (100%)	0.065
PD-L1 inhibitor	11 (31%)	1 (33%)	10 (42%)	0 (0%)	0.065
CTLA4 inhibitor	5 (14%)	0 (0%)	2 (8.3%)	3 (38%)	0.11
Myocarditis	13 (37%)	0 (0%)	13 (54%)	0 (0%)	**0.006**
Diplopia	7 (20%)	0 (0%)	6 (25%)	1 (12%)	0.8
Dysphagia	4 (11%)	0 (0%)	2 (8.3%)	2 (25%)	0.5
Anti-striational	15 (43%)	0 (0%)	11 (46%)	4 (50%)	0.6
Anti-AChR	14 (40%)	0 (0%)	12 (50%)	2 (25%)	**0.010**
Other myositis autoantibodies	16 (46%)	3 (100%)	12 (50%)	1 (12%)	0.051
CK at biopsy	652 (219, 1,218)	428 (278, 4,758)	896 (278, 1,394)	427 (62, 830)	0.3
Peak CK	1,184 (462, 6,118)	428 (278, 4,758)	1,240 (899, 6,565)	652 (379, 4,771)	0.4

[1]n (%); Median (IQR)
[2]Fisher's exact test; Kruskal-Wallis rank sum test

Figure 2. Histological appearance of patients with immune-checkpoint induced myopathy from clusters ICI-DM (A-D), ICI-MYO1 (F-I), and ICI-MYO2 (J-M). ICI-DM biopsies showed perifascicular atrophy, and intense vascular damage (star marks an area of muscle infarction). ICI-MYO1 showed intense inflammatory infiltrates with abundant macrophages and T-cells. Finall, ICI-MYO2 had predominant necrosis (white arrow indicates an area of myophagocytosis) with few inflammatory cells. The first column shows H&E (A, F, J), the second MHC-1 (B, G, K), the third MHC-2 (C, H, L). The fourth column shows CD68 in I and MX1 in D and M.

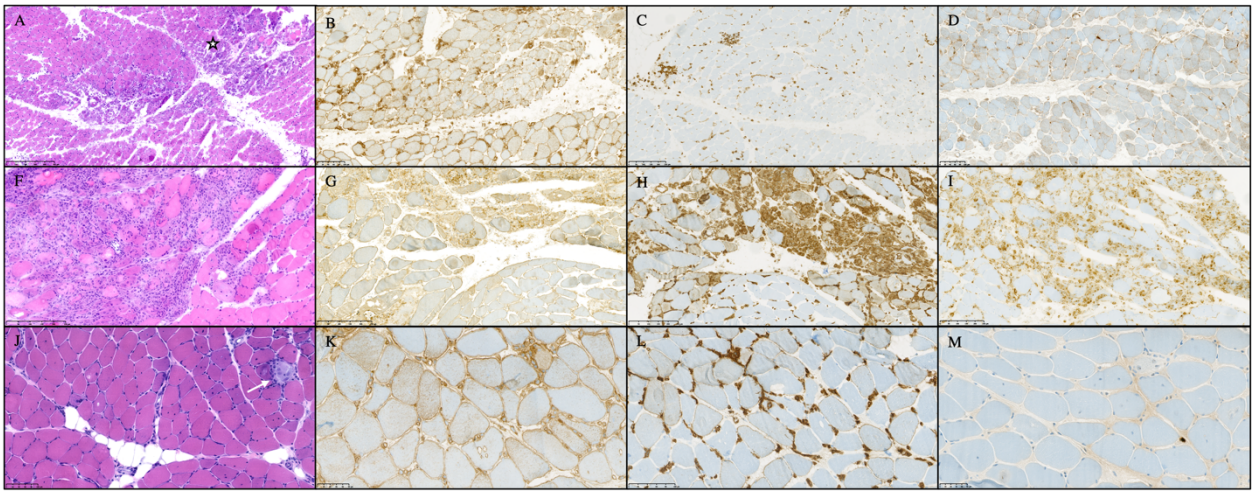
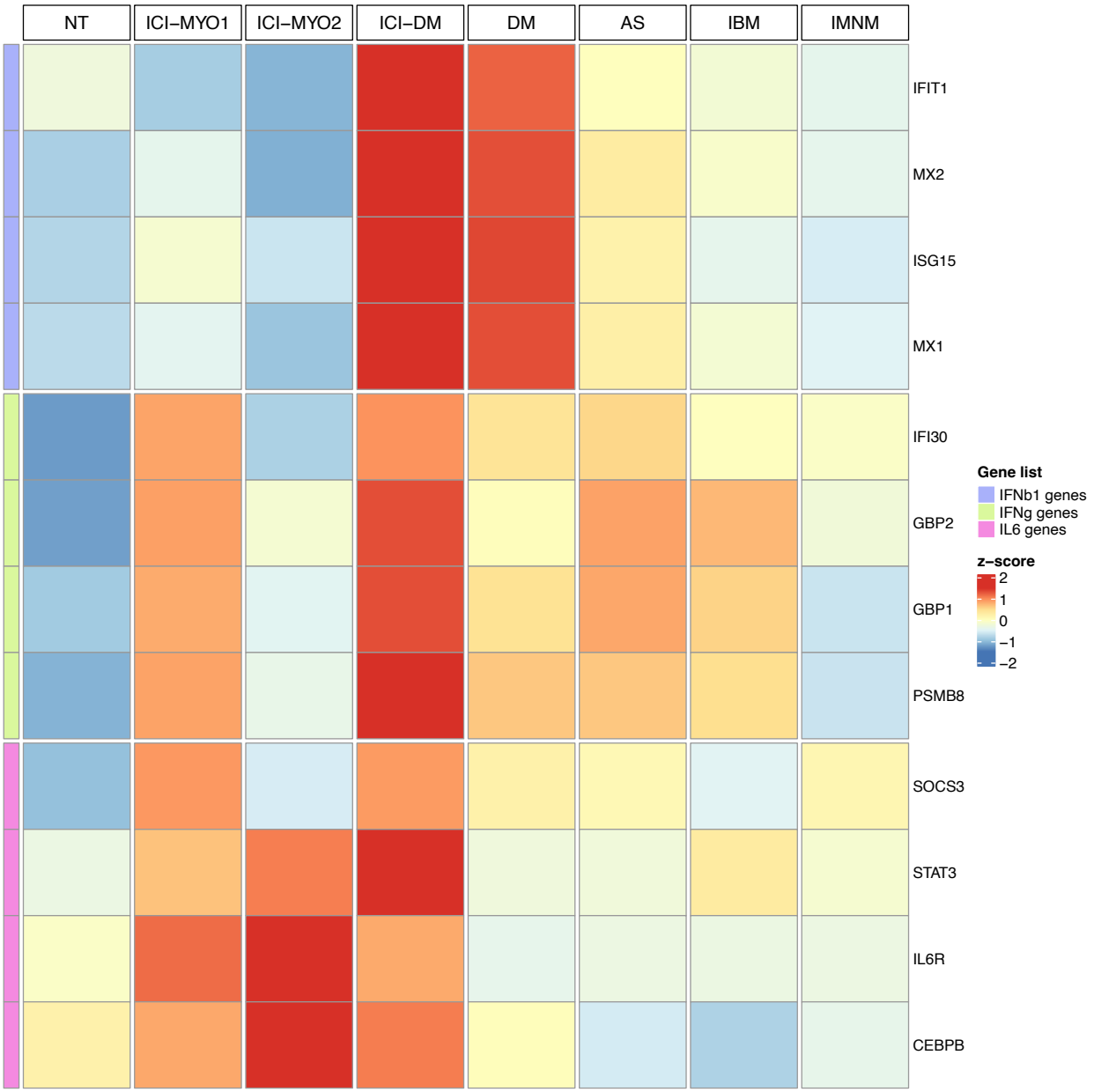


Figure 3. Expression (average z-score of log2[TMM+1]) of IFNB1, IFNG, and IL6 related genes in the three clusters of patients with ICI-induced myopathy (ICI-MYO1, ICI-MYO2, and ICI-DM) and in the comparator muscles biopsies.



NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy. These IFNB1 and IFNG inducible genes were validated in cultures of differentiating human skeletal muscle myoblasts treated with IFNB1 and IFNG.

Figure 4. Top pathways in gene set enrichment analysis using the Reactome database in the three clusters of patients with ICI-induced myopathy (ICI-DM [left], ICI-MYO1 [middle], and ICI-MYO2 [right]).

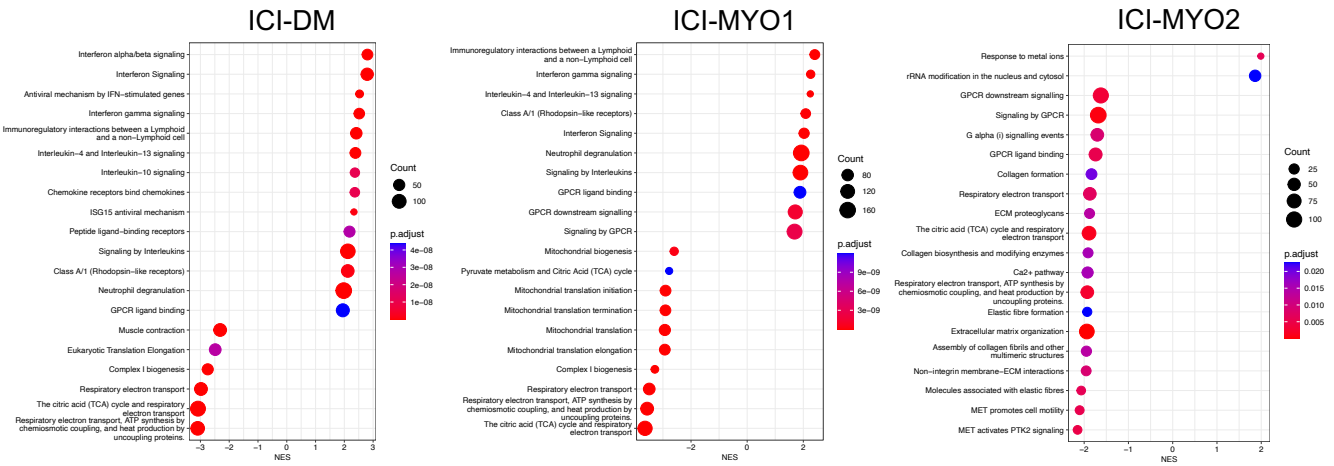


Figure 5. Percent of cells expressing genes related to the type 1 (IFIT1, MX2, ISG15, MX1) and type 2 (IFNG, IFI30, GBP2, GBP1, PSMB8) interferon pathway in muscle biopsies from clusters ICI-DM, and ICI-MYO1 of immune checkpoint-induced myopathy compared with a representative selection of patients with other types of inflammatory myopathy (4 dermatomyositis, 3 antisynthetase syndrome, 6 immune-mediated necrotizing myositis, and 2 inclusion body myositis).

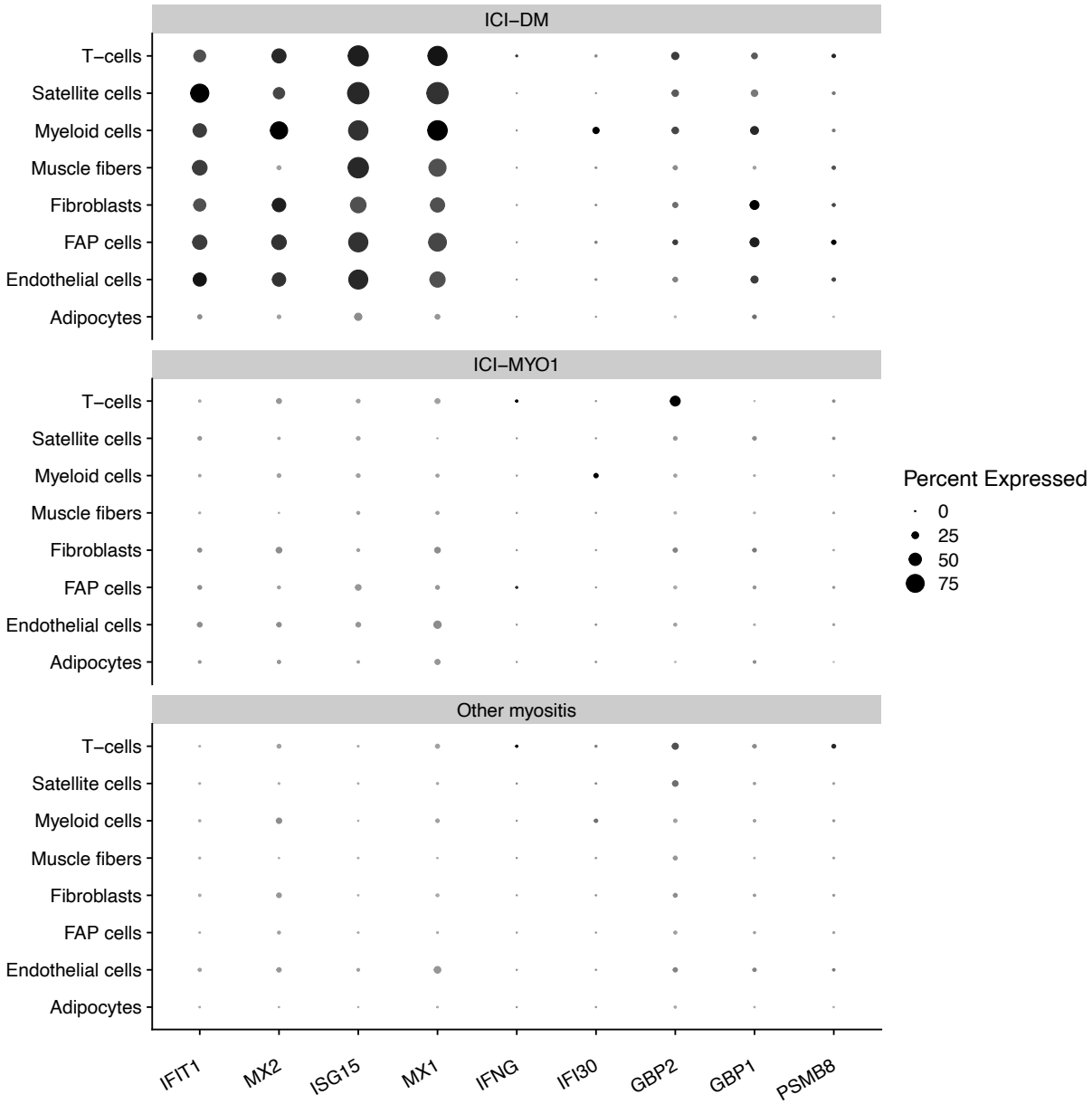


Figure 6. Percent of cells expressing genes related to the IL6 pathway in muscle biopsies from immune checkpoint-induced myopathy from clusters ICI-DM, and ICI-MYO1 compared with a representative selection of patients with other types of inflammatory myopathy (4 dermatomyositis, 3 antisynthetase syndrome, 6 immune-mediated necrotizing myositis, and 2 inclusion body myositis).

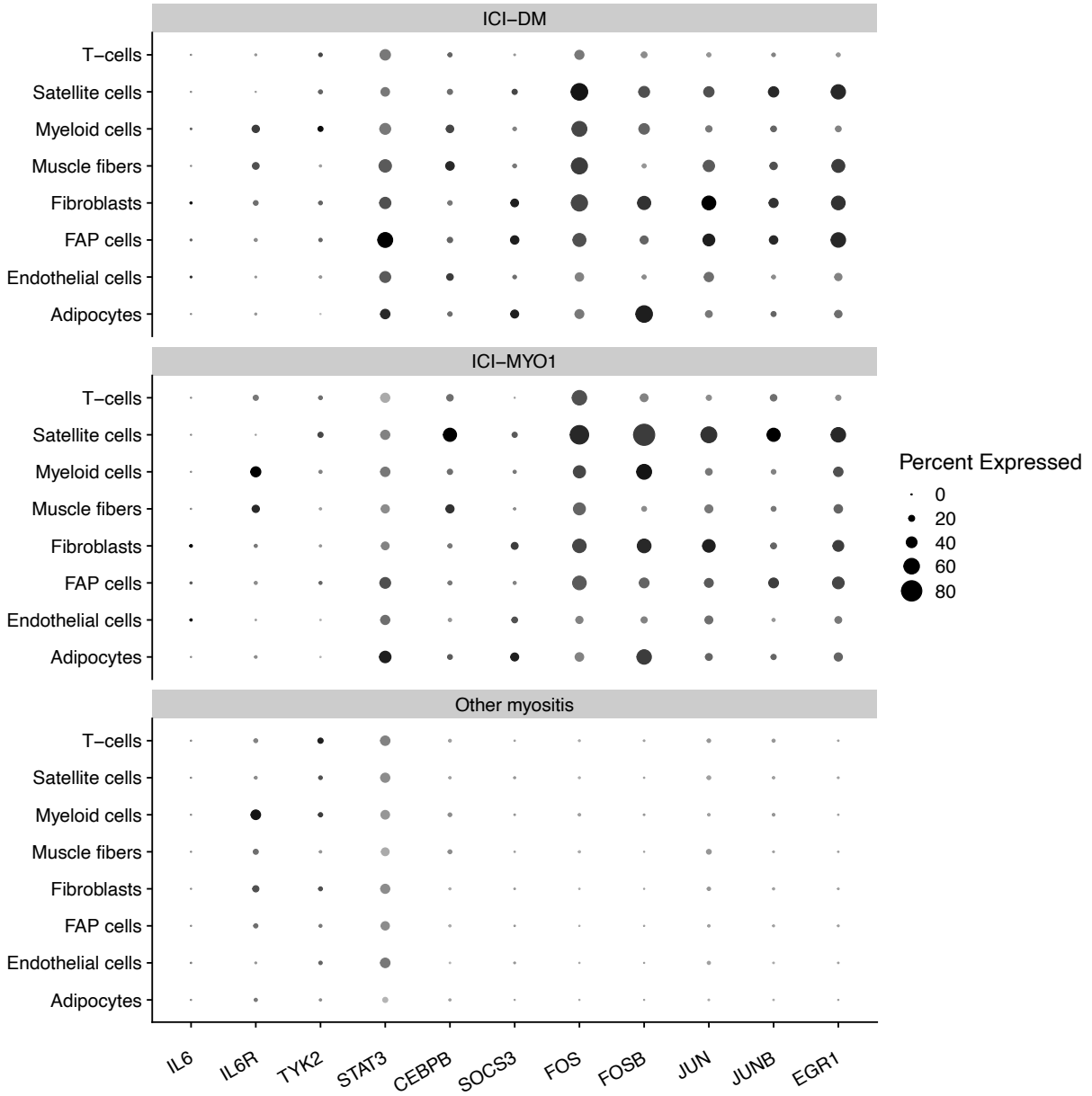
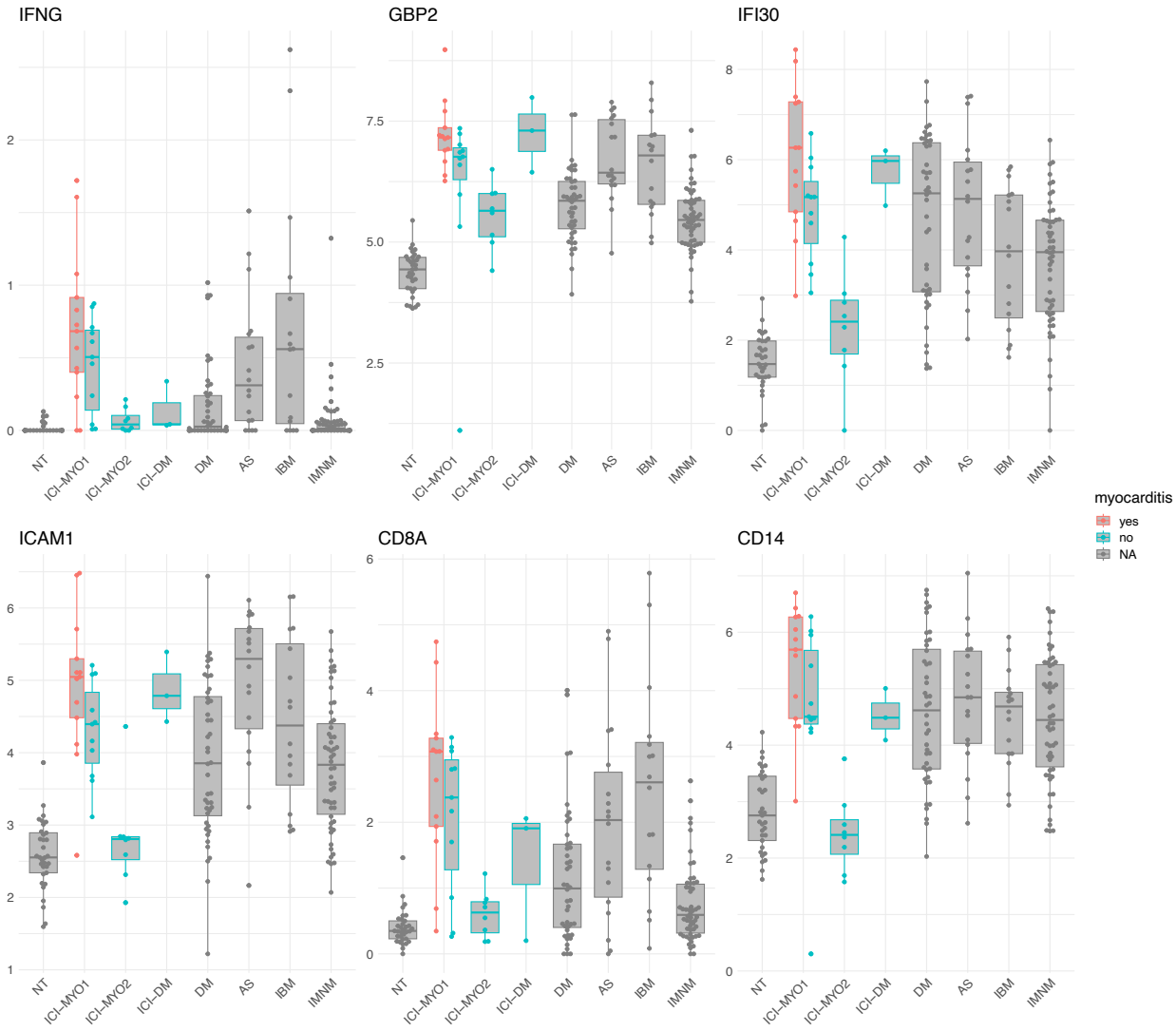


Figure 7. Expression levels (log2[TMM + 1]) of IFNG, IFNG-inducible genes, ICAM1, CD8A, and CD14 in patients according to the presence of myocarditis.



NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Bibliography

1. Selva-O'Callaghan A, Pinal-Fernandez I, Trallero-Araguas E, Milisenda JC, Grau-Junyent JM, Mammen AL. Classification and management of adult inflammatory myopathies. Lancet Neurol. 2018 Sep; 17(9):816-828.

2. Pinal-Fernandez I, Casal-Dominguez M, Derfoul A, Pak K, Miller FW, Milisenda JC, et al. Machine learning algorithms reveal unique gene expression profiles in muscle biopsies from patients with different types of myositis. Ann Rheum Dis. 2020 Sep; 79(9):1234-1242.

3. Johnson DB, Balko JM, Compton ML, Chalkias S, Gorham J, Xu Y, et al. Fulminant Myocarditis with Combination Immune Checkpoint Blockade. N Engl J Med. 2016 Nov 3; 375(18):1749-1755.

4. Vermeulen L, Depuydt CE, Weckx P, Bechter O, Van Damme P, Thal DR, et al. Myositis as a neuromuscular complication of immune checkpoint inhibitors. Acta Neurol Belg. 2020 Apr; 120(2):355-364.

5. Matas-Garcia A, Milisenda JC, Selva-O'Callaghan A, Prieto-Gonzalez S, Padrosa J, Cabrera C, et al. Emerging PD-1 and PD-1L inhibitors-associated myopathy with a characteristic histopathological pattern. Autoimmun Rev. 2020 Feb; 19(2):102455.

6. Touat M, Maisonobe T, Knauss S, Ben Hadj Salem O, Hervier B, Aure K, et al. Immune checkpoint inhibitor-related myositis and myocarditis in patients with cancer. Neurology. 2018 Sep 4; 91(10):e985-e994.

7. Shelly S, Triplett JD, Pinto MV, Milone M, Diehn FE, Zekeridou A, et al. Immune checkpoint inhibitor-associated myopathy: a clinicoseropathologically distinct myopathy. Brain Commun. 2020; 2(2):fcaa181.

8. Amici DR, Pinal-Fernandez I, Mazala DA, Lloyd TE, Corse AM, Christopher-Stine L, et al. Calcium dysregulation, functional calpainopathy, and endoplasmic reticulum stress in sporadic inclusion body myositis. Acta Neuropathol Commun. 2017 Mar 22; 5(1):24.

9. Pinal-Fernandez I, Amici DR, Parks CA, Derfoul A, Casal-Dominguez M, Pak K, et al. Myositis Autoantigen Expression Correlates With Muscle Regeneration but Not Autoantibody Specificity. Arthritis Rheumatol. 2019 Aug; 71(8):1371-1376.

10. Pinal-Fernandez I, Casal-Dominguez M, Derfoul A, Pak K, Plotz P, Miller FW, et al. Identification of distinctive interferon gene signatures in different types of myositis. Neurology. 2019 Sep 17; 93(12):e1193-e1204.

11. Amici DR, Pinal-Fernandez I, Christopher-Stine L, Mammen AL, Mendillo ML. A network of core and subtype-specific gene expression programs in myositis. Acta Neuropathol. 2021 Nov; 142(5):887-898.

12. Riaz N, Havel JJ, Makarov V, Desrichard A, Urba WJ, Sims JS, et al. Tumor and Microenvironment Evolution during Immunotherapy with Nivolumab. Cell. 2017 Nov 2; 171(4):934-949 e916.

13. Schirmer L, Velmeshev D, Holmqvist S, Kaufmann M, Werneburg S, Jung D, et al. Neuronal vulnerability and multilineage diversity in multiple sclerosis. Nature. 2019 Sep; 573(7772):75-82.

14. Tibshirani R, Walther G. Estimating the number of clusters in a dataset via the Gap statistic. R Statist Soc. 2000; 63.

15. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A. 2005 Oct 25; 102(43):15545-15550.

16. Croker BA, Krebs DL, Zhang JG, Wormald S, Willson TA, Stanley EG, et al. SOCS3 negatively regulates IL-6 signaling in vivo. Nat Immunol. 2003 Jun; 4(6):540-545.

17. Xiao W, Hodge DR, Wang L, Yang X, Zhang X, Farrar WL. NF-kappaB activates IL-6 expression through cooperation with c-Jun and IL6-AP1 site, but is independent of its IL6-NFkappaB regulatory site in autocrine human multiple myeloma cells. Cancer Biol Ther. 2004 Oct; 3(10):1007-1017.

18. Hsu W, Kerppola TK, Chen PL, Curran T, Chen-Kiang S. Fos and Jun repress transcription activation by NF-IL6 through association at the basic zipper region. Mol Cell Biol. 1994 Jan; 14(1):268-276.

19. Luo Y, Zheng SG. Hall of Fame among Pro-inflammatory Cytokines: Interleukin-6 Gene and Its Transcriptional Regulation Mechanisms. Front Immunol. 2016; 7:604.

20. Baccam M, Woo SY, Vinson C, Bishop GA. CD40-mediated transcriptional regulation of the IL-6 gene in B lymphocytes: involvement of NF-kappa B, AP-1, and C/EBP. J Immunol. 2003 Mar 15; 170(6):3099-3108.

21. Mammen AL, Rajan A, Pak K, Lehky T, Casciola-Rosen L, Donahue RN, et al. Pre-existing antiacetylcholine receptor autoantibodies and B cell lymphopaenia are associated with the development of myositis in patients with thymoma treated with avelumab, an immune checkpoint inhibitor targeting programmed death-ligand 1. Ann Rheum Dis. 2019 Jan; 78(1):150-152.

22. Pinal-Fernandez I, Ferrer-Fabregas B, Trallero-Araguas E, Balada E, Martinez MA, Milisenda JC, et al. Tumour TIF1 mutations and loss of heterozygosity related to cancer-associated myositis. Rheumatology (Oxford). 2018 Feb 1; 57(2):388-396.

23. Cordel N, Derambure C, Coutant S, Mariette X, Jullien D, Debarbieux S, et al. TRIM33 gene somatic mutations identified by next generation sequencing in neoplasms of patients with anti-TIF1gamma positive cancer-associated dermatomyositis. Rheumatology (Oxford). 2021 Dec 1; 60(12):5863-5867.

24. Cheng W, Sun T, Liu C, Zhou Z, Duan J, Zhao Y, et al. A systematic review of myasthenia gravis complicated with myocarditis. Brain Behav. 2021 Aug; 11(8):e2242.

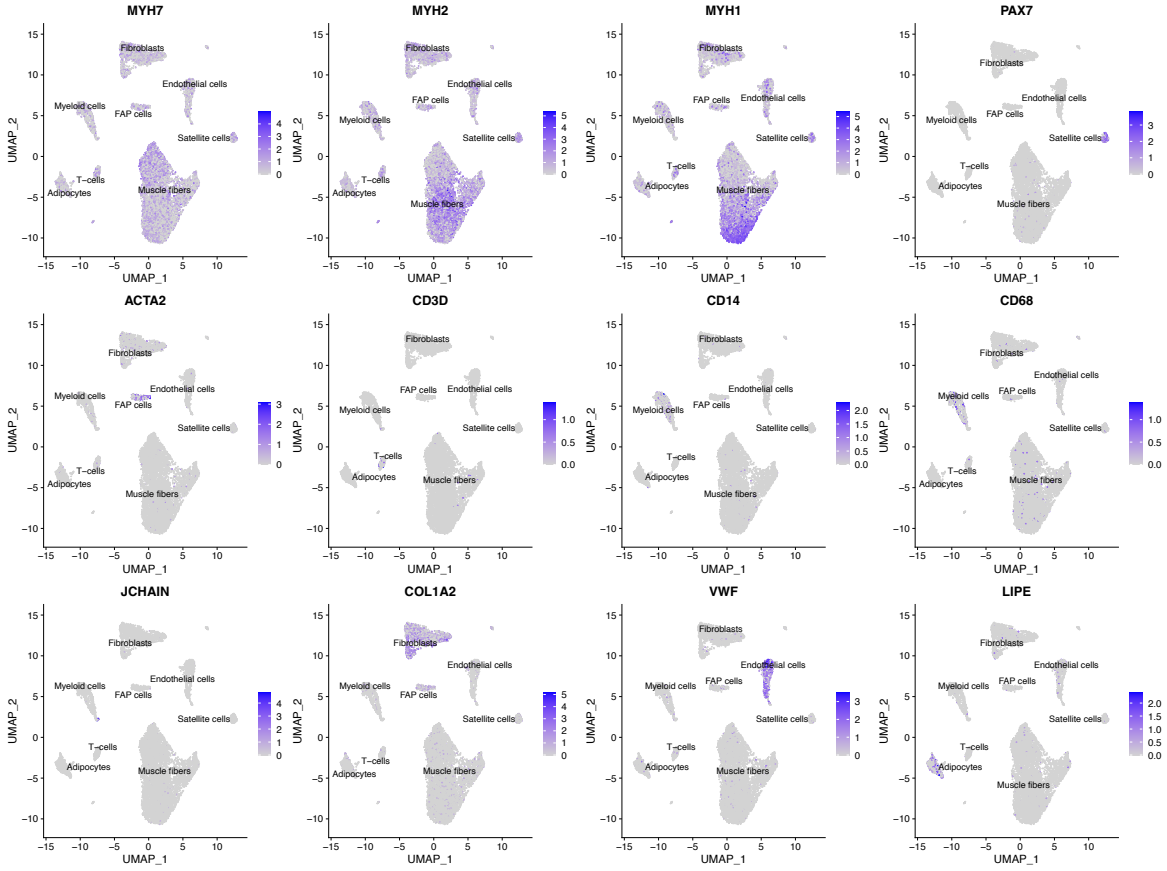
25. Diamanti L, Picca A, Bini P, Gastaldi M, Alfonsi E, Pichiecchio A, et al. Characterization and management of neurological adverse events during immune-checkpoint inhibitors treatment: an Italian multicentric experience. Neurol Sci. 2022 Mar; 43(3):2031-2041.

26. Salem JE, Allenbach Y, Vozy A, Brechot N, Johnson DB, Moslehi JJ, et al. Abatacept for Severe Immune Checkpoint Inhibitor-Associated Myocarditis. N Engl J Med. 2019 Jun 13; 380(24):2377-2379.

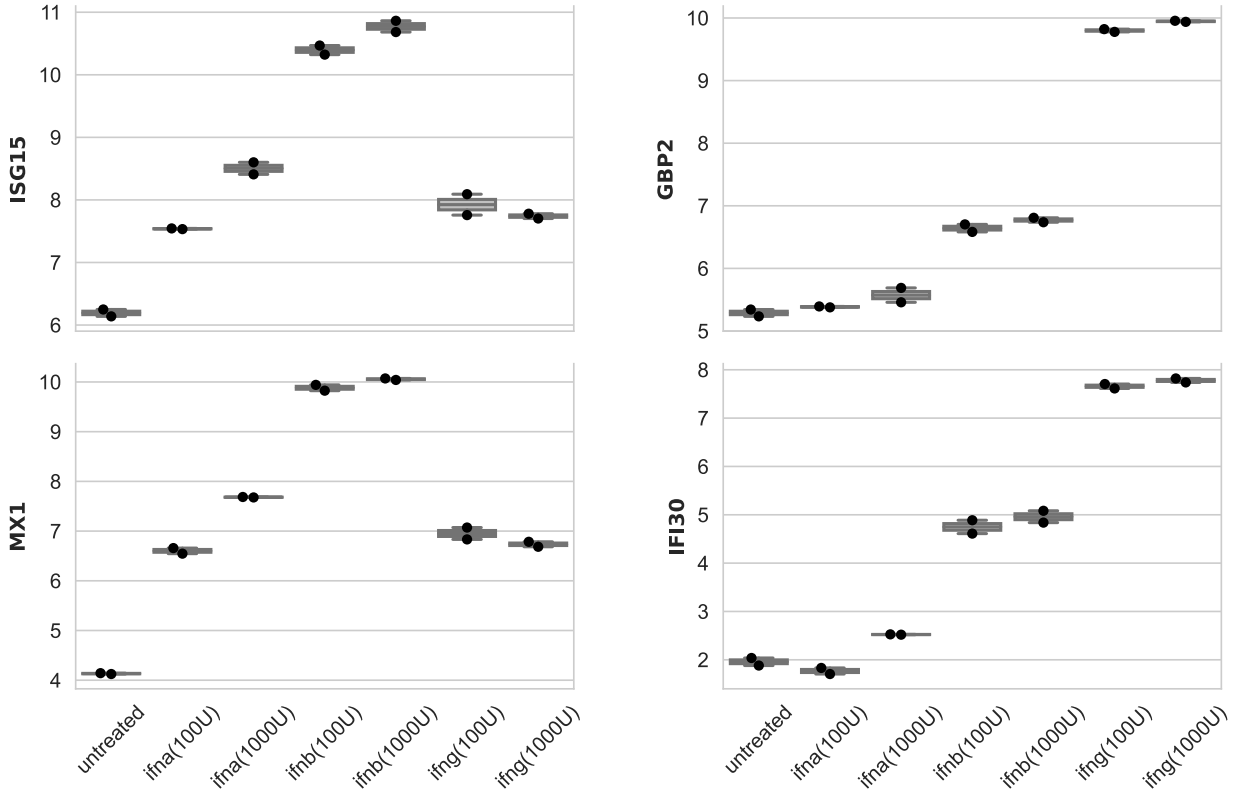
27. Hailemichael Y, Johnson DH, Abdel-Wahab N, Foo WC, Bentebibel SE, Daher M, et al. Interleukin-6 blockade abrogates immunotherapy toxicity and promotes tumor immunity. Cancer Cell. 2022 May 9; 40(5):509-523 e506.

28. Kim ST, Tayar J, Trinh VA, Suarez-Almazor M, Garcia S, Hwu P, et al. Successful treatment of arthritis induced by checkpoint inhibitors with tocilizumab: a case series. Ann Rheum Dis. 2017 Dec; 76(12):2061-2064.

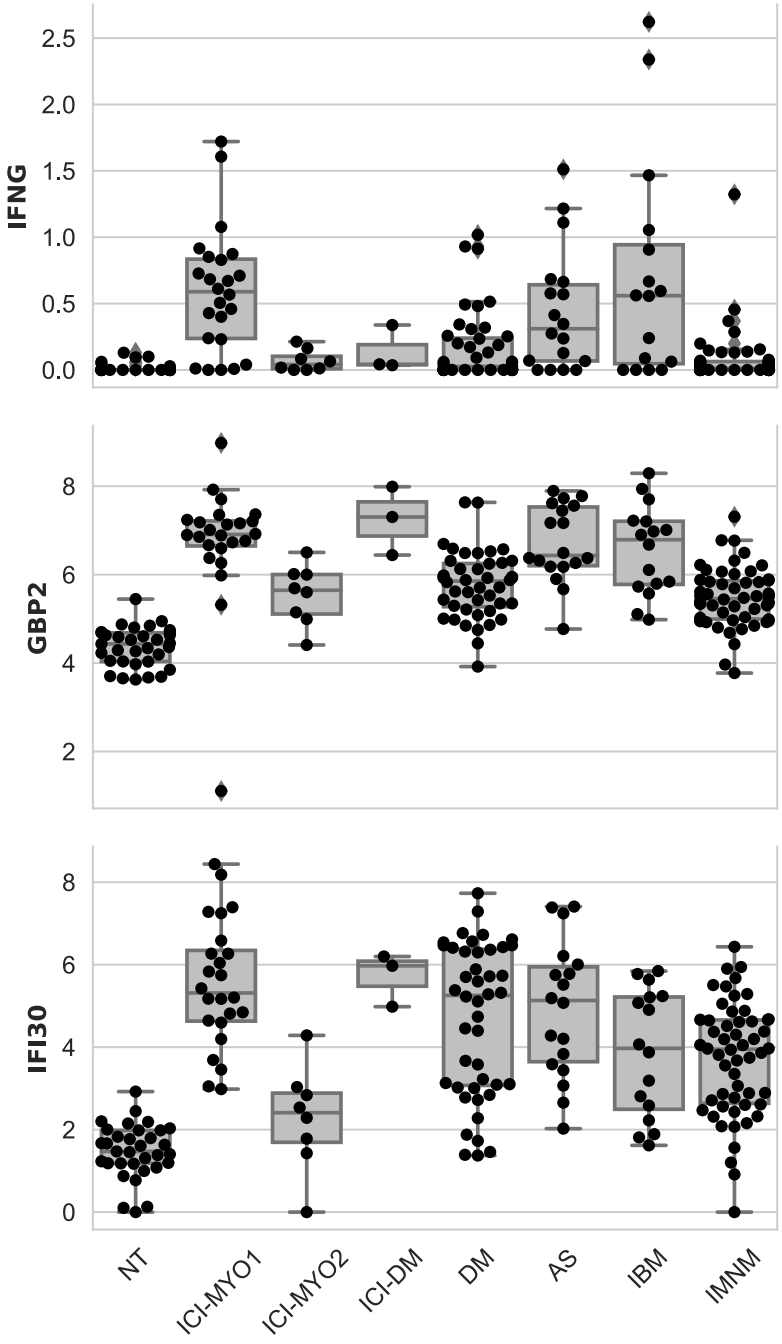
Supplementary Figure 1. Representative genes for each single-nuclei RNA sequencing cluster.



Supplementary Figure 2. Expression (log2[TMM+1]) of predominantly type1 interferon-stimulated genes (ISG15, and MX1), and predominantly type 2 interferon-stimulated genes (GBP2, and IFI30) in differentiating human skeletal muscle myoblasts treated with IFNA2a, IFNB1, and IFNG at two different doses each (100U, and 1000U).

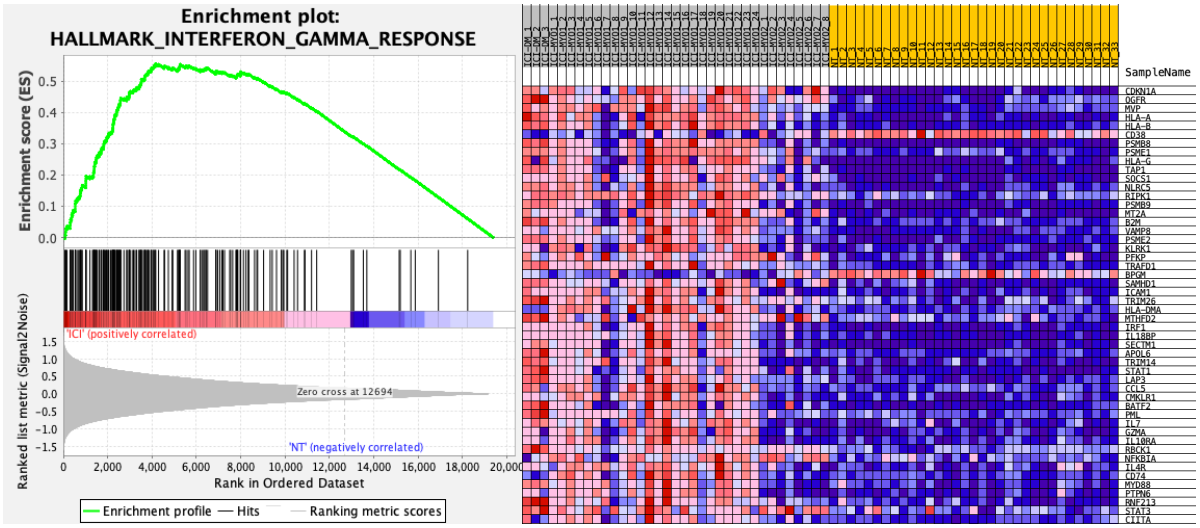


Supplementary Figure 3. Expression (log2[TMM+1]) of IFNG and representative IFNG-stimulated genes.

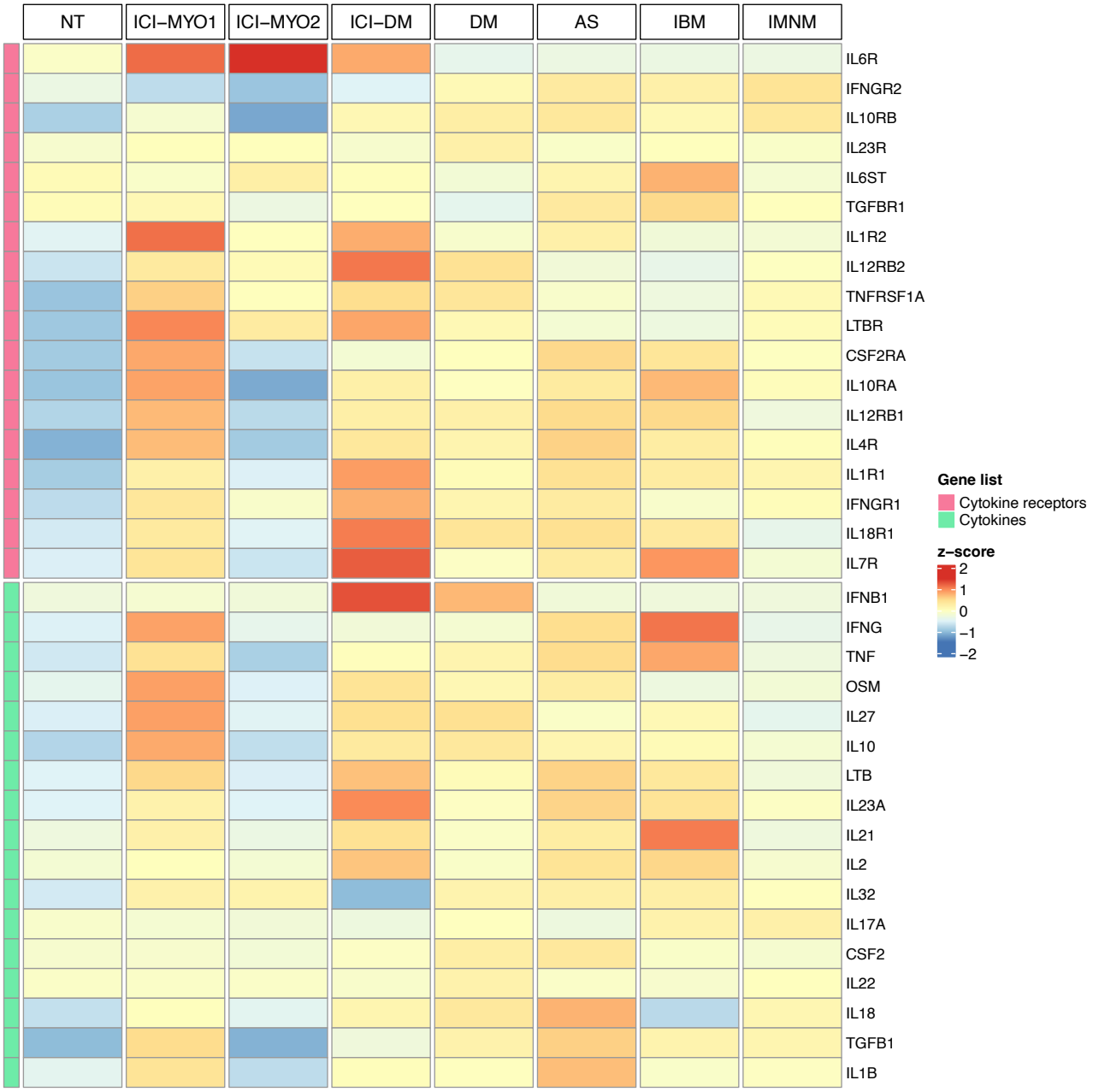


NT: normal muscle; ICI-DM: immune checkpoint-induced dermatomyositis; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 4. Gene Set Enrichment Analysis of the interferon-gamma response (left) in immune checkpoint-induced myopathy patients compared to normal muscle (p-value 0.04). Fifty genes with the highest signal-to-noise ratio in this pathway (red high, blue low).

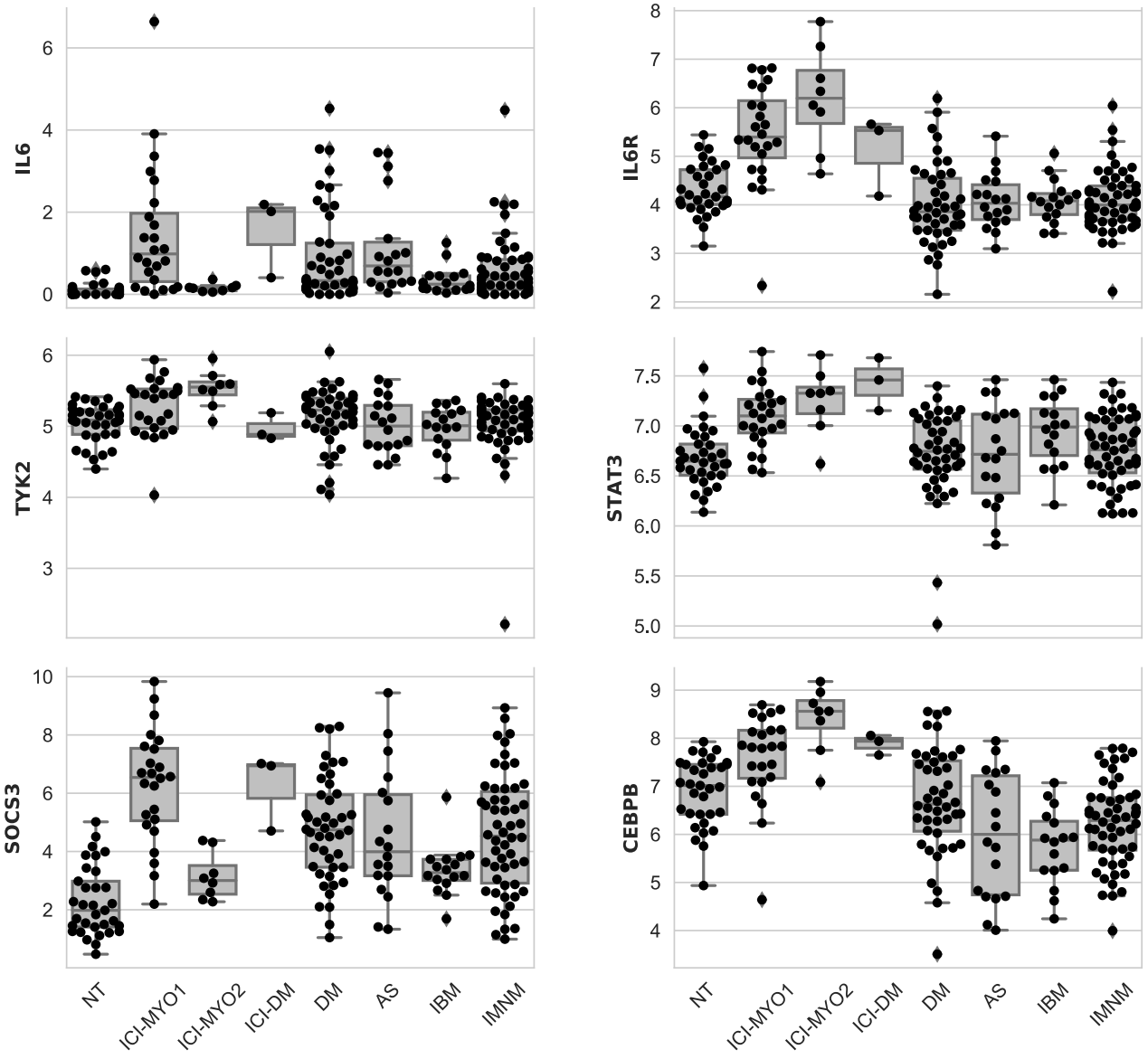


Supplementary Figure 5. Expression (average z-score of log2[TMM+1]) of cytokines and cytokine receptors in patients with ICI-induced myopathy (ICI-MYO1, ICI-MYO2, and ICI-DM) and in the comparator muscles biopsies.



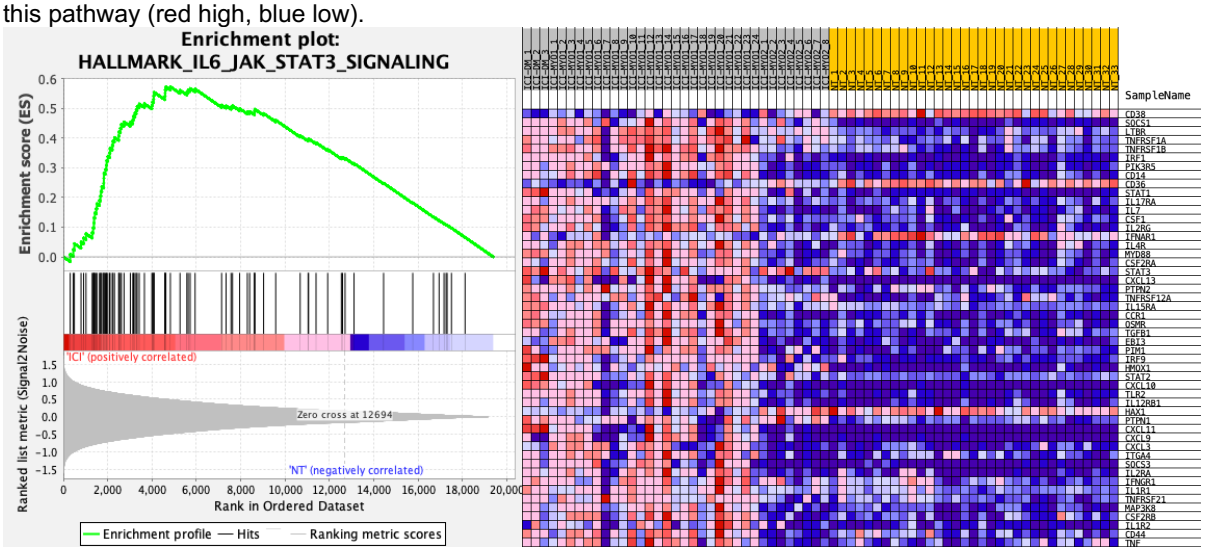
NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 6. Expression (log2[TMM+1]) of representative genes from the IL6 pathway.

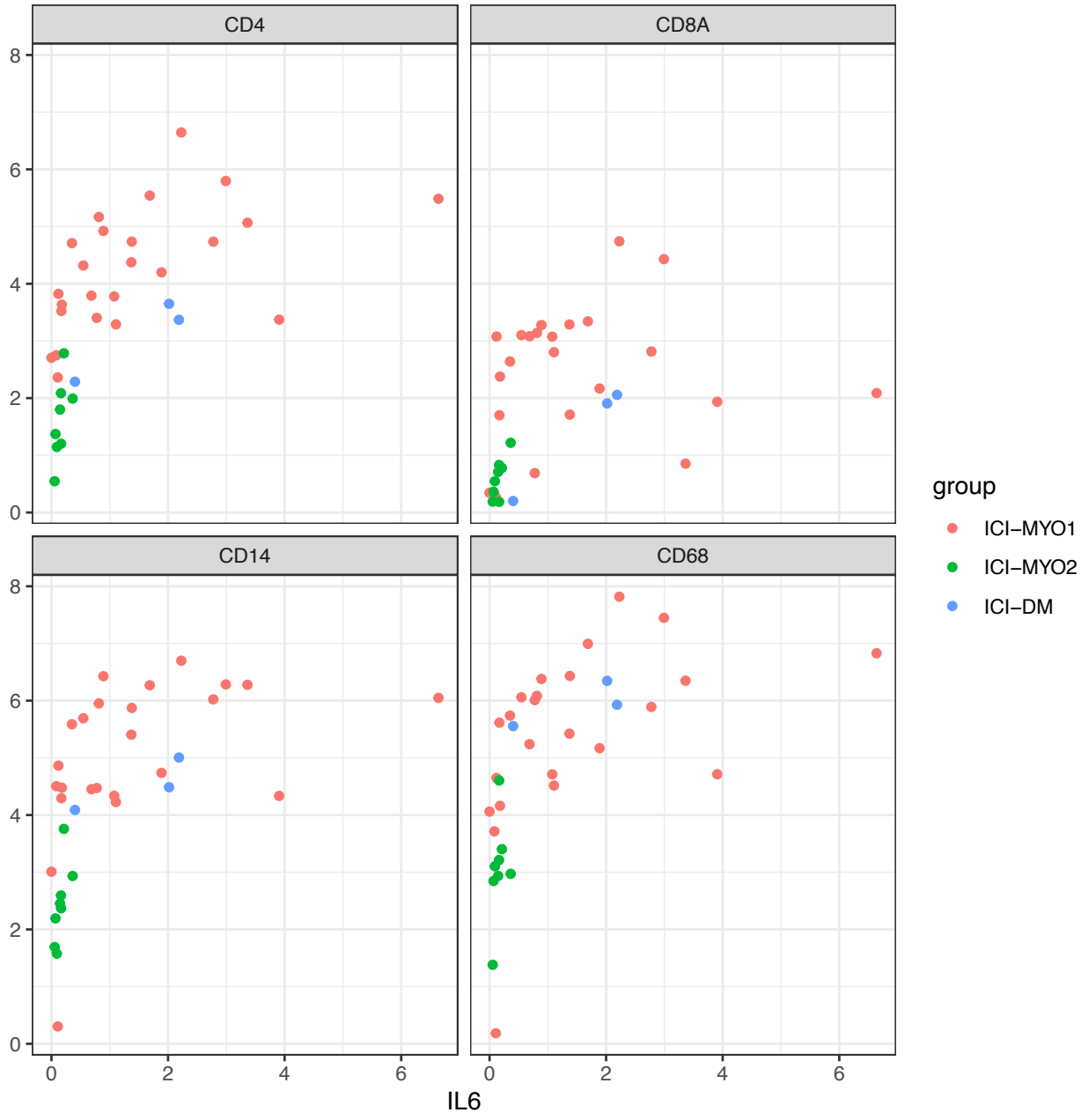


NT: normal muscle; ICI-DM: immune checkpoint-induced dermatomyositis; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

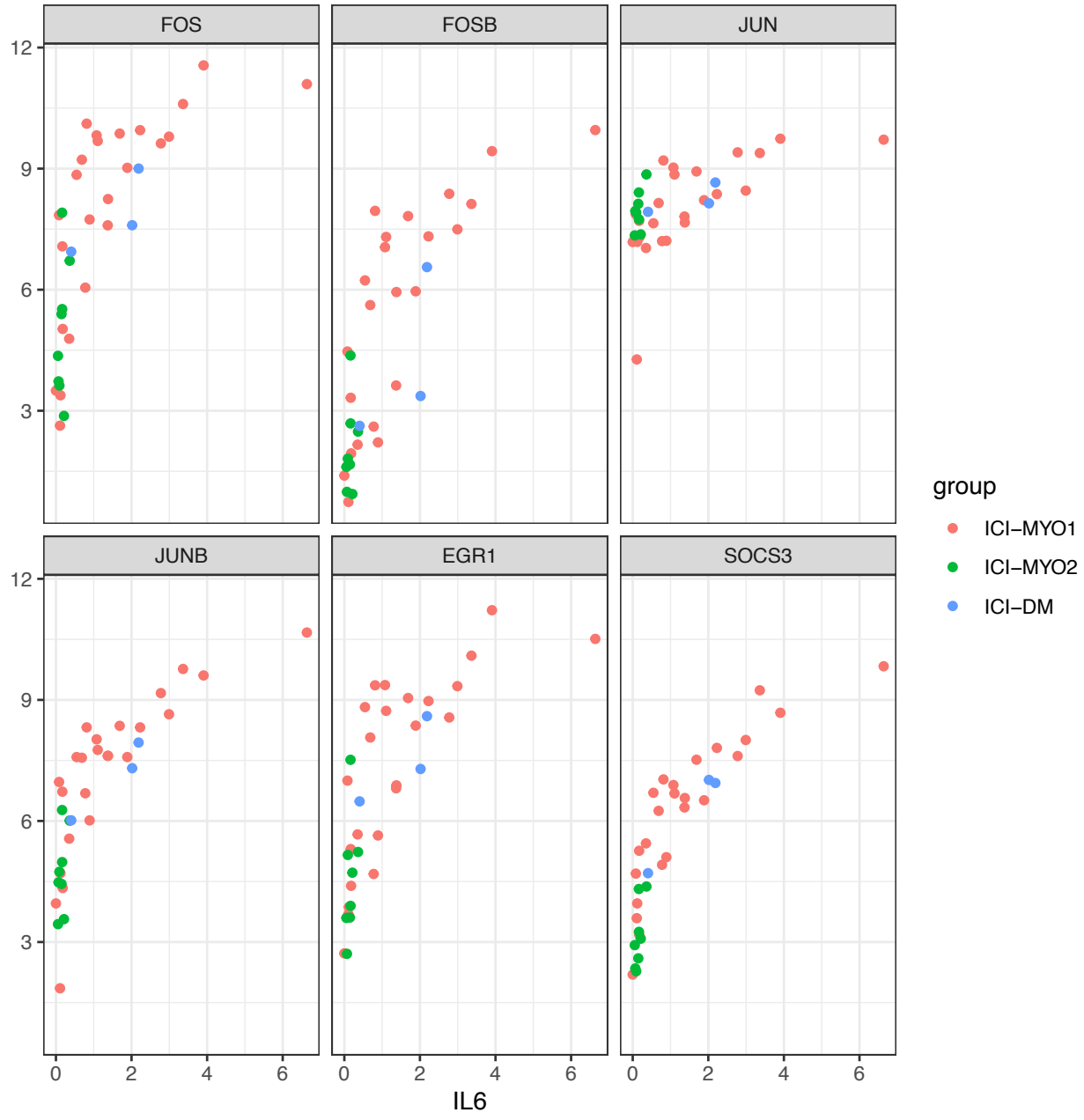
Supplementary Figure 7. Gene Set Enrichment Analysis of the IL6-JAK-STAT3 pathway (left) in immune checkpoint-induced myopathy patients compared to normal muscle (p-value 0.01). Fifty genes with the highest signal-to-noise ratio in this pathway (red high, blue low).



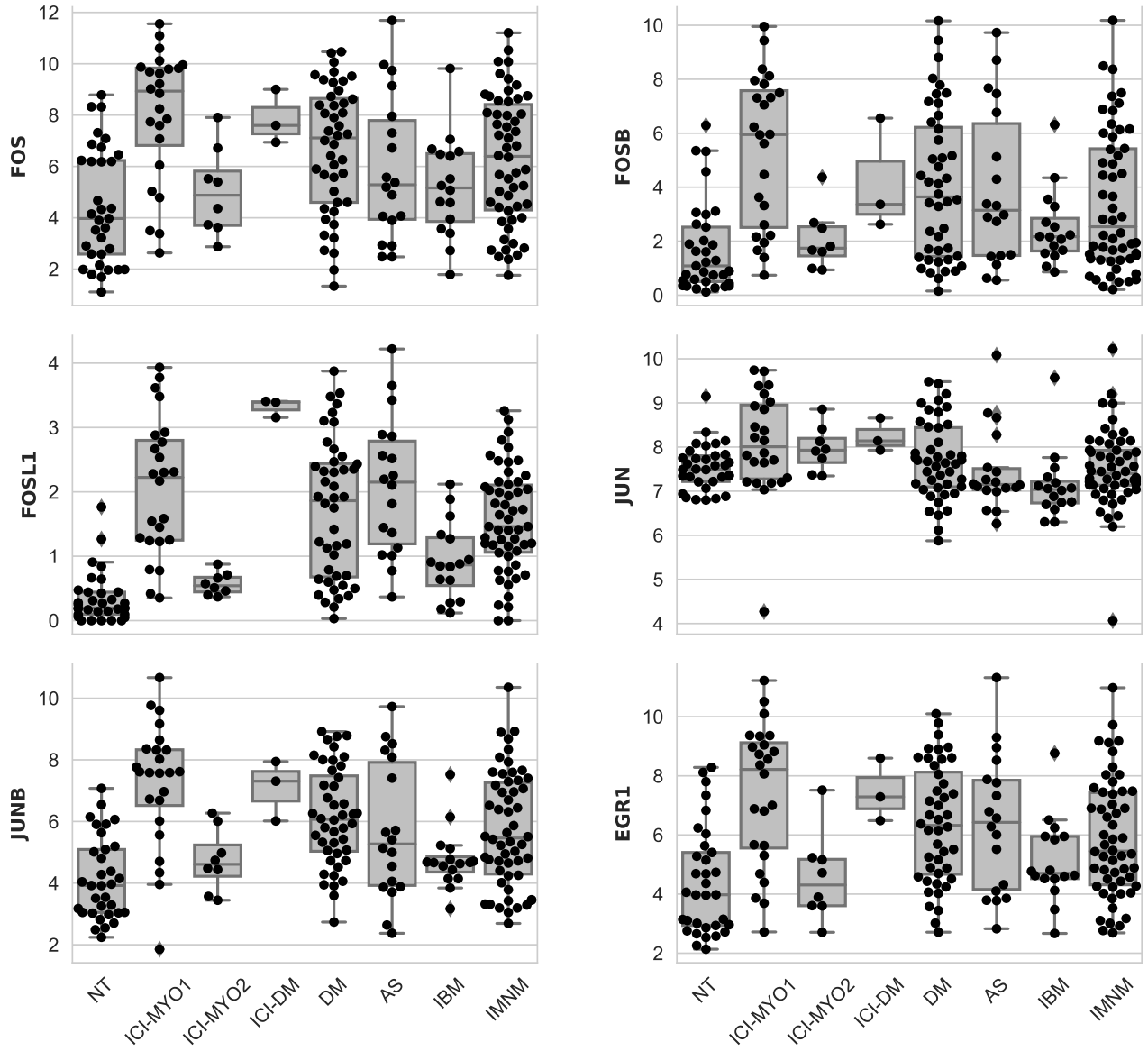
Supplementary Figure 8. Correlation of CD4, CD8A, CD14, and CD68 with IL6 in the different clusters of patients with immune checkpoint-induced myopathy (ICI-MYO1, ICI-MYO2, and ICI-DM)



Supplementary Figure 9. Correlation of IL6 with EGR1, SOCS3, and members of the protein families FOS and JUN in the different clusters of patients with immune checkpoint-induced myopathy (ICI-MYO1, ICI-MYO2, and ICI-DM)

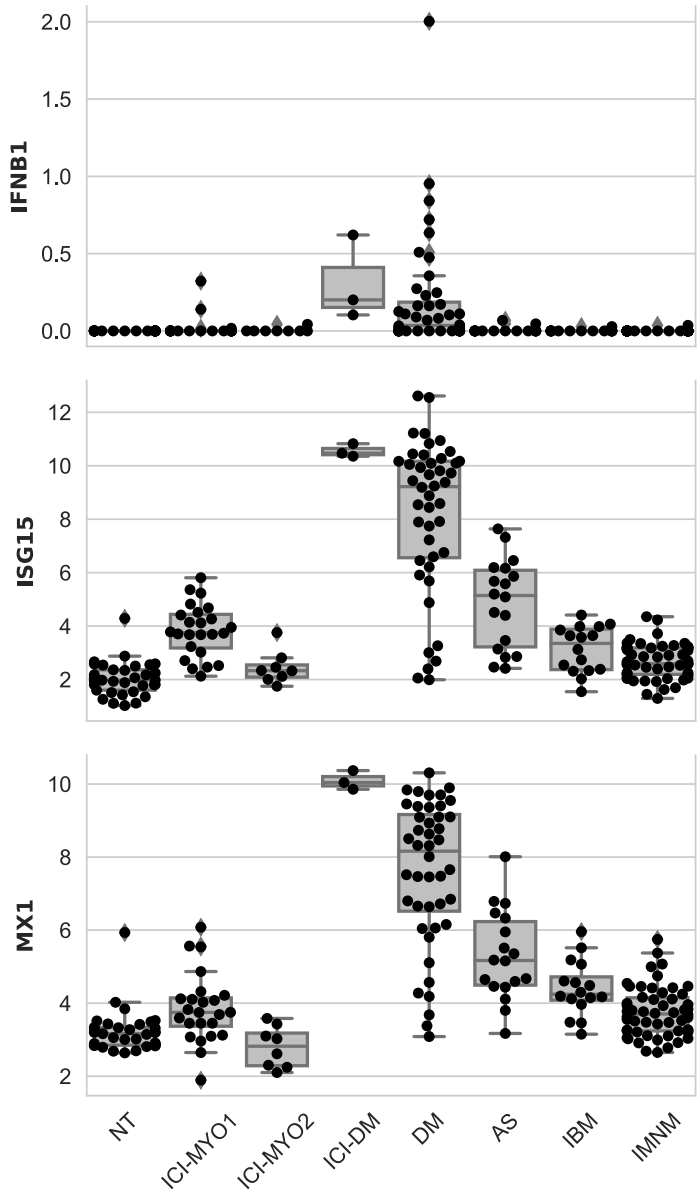


Supplementary Figure 10. Expression (log2[TMM+1]) of EGR1 and members of the FOS and JUN family of proteins.



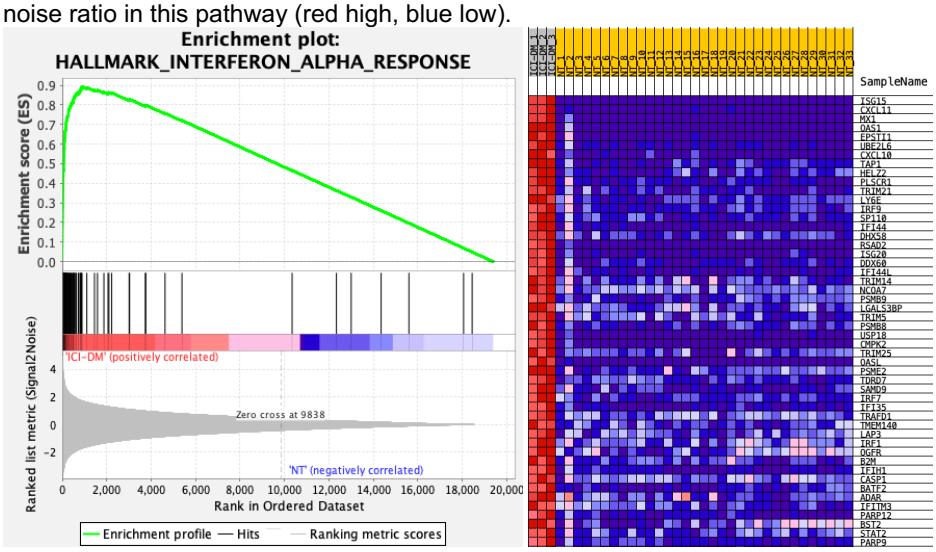
NT: normal muscle; ICI-DM: immune checkpoint-induced dermatomyositis; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Figure 11. Expression levels (log2[TMM + 1]) of IFNB1, and interferon type I inducible genes ISG15 and MX1 in muscle. In patients with immune-checkpoint inhibitor-induced dermatomyositis (ICI-DM), IFNB1 and IFN type I inducible genes are overexpressed, similar to patients with non-ICI dermatomyositis.

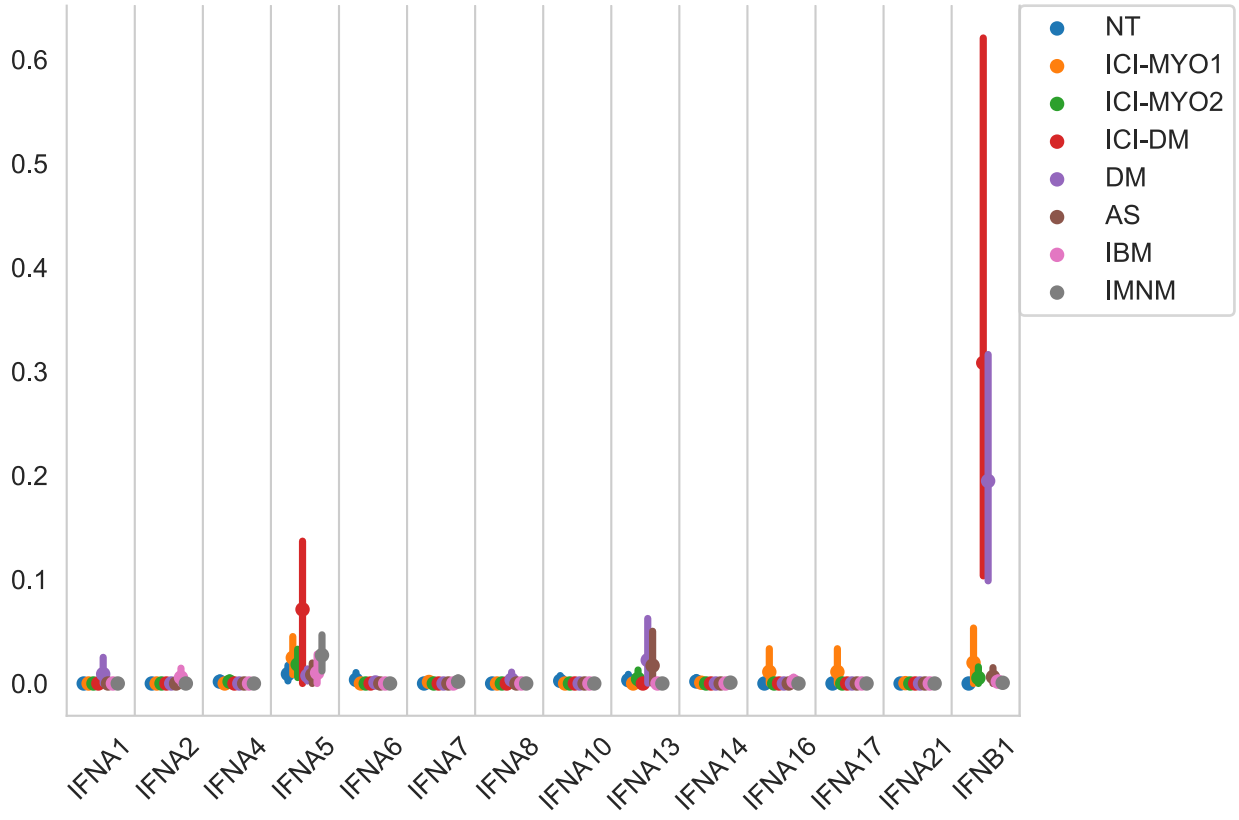


NT: normal muscle; ICI-DM: immune checkpoint-induced dermatomyositis; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 12. Gene Set Enrichment Analysis of the type 1 interferon pathway (left) in immune checkpoint-induced dermatomyositis patients compared to normal muscle (p-value < 0.001). Fifty genes with the highest signal-to-noise ratio in this pathway (red high, blue low).

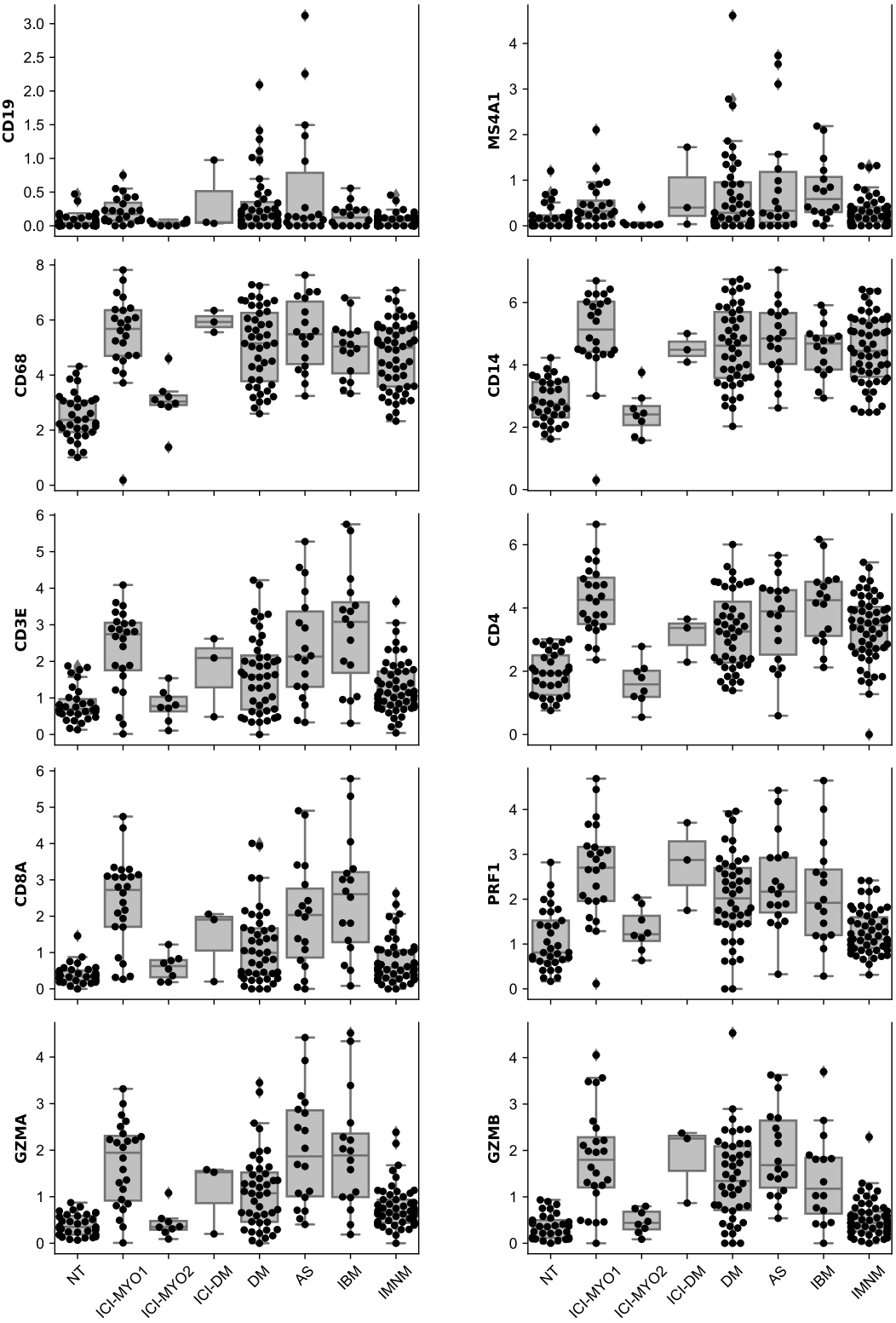


Supplementary Figure 13. Average expression levels (average and 95% confidence interval of log2[TMM + 1]) of type I interferon genes in muscle. Patients with immune checkpoint dermatomyositis have levels of IFNB1 similar to patients with dermatomyositis.



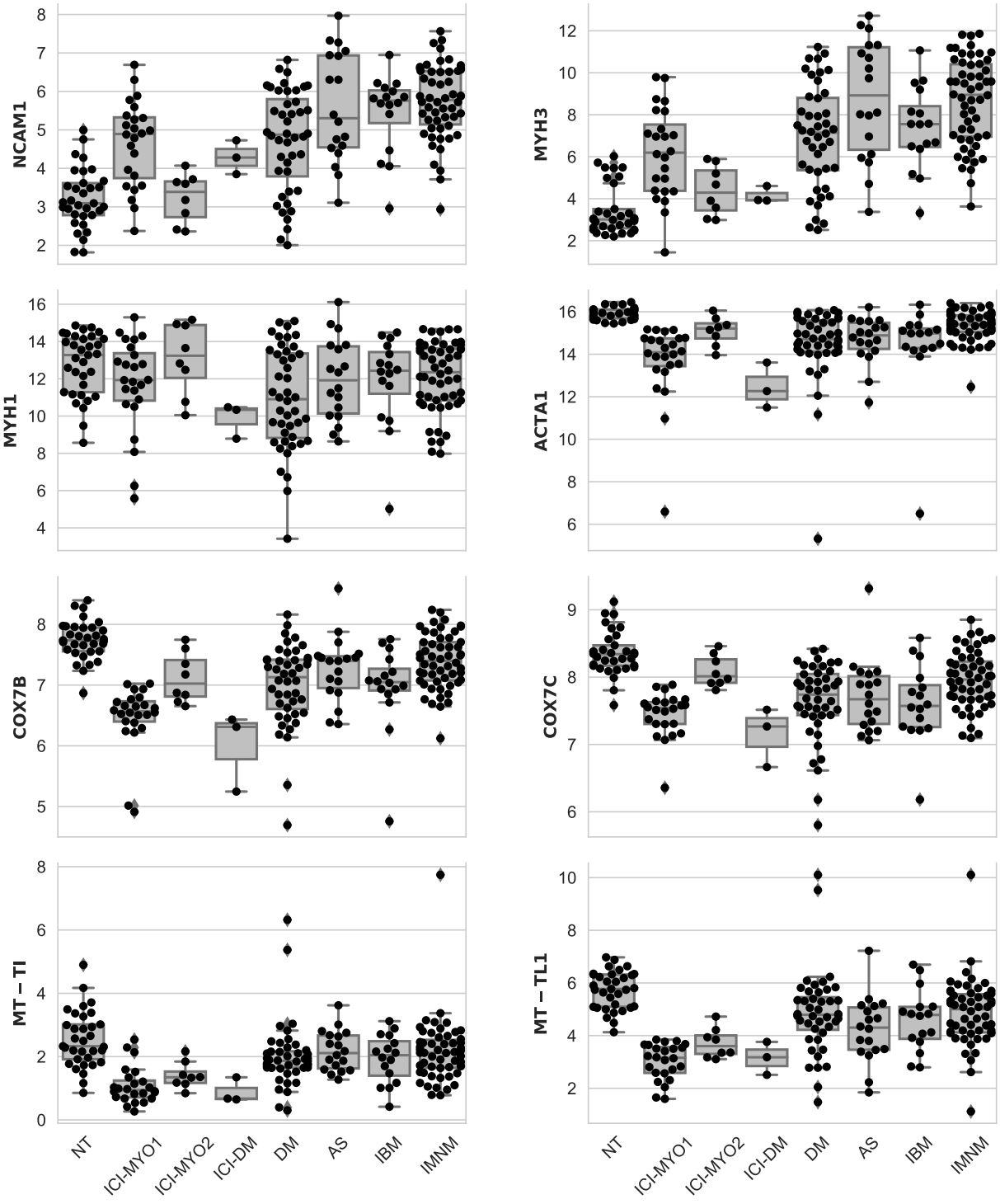
NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy

Supplementary Figure 14. Expression (log2[TMM+1]) of representative gene markers associated with B cells, macrophages, and T-cells.



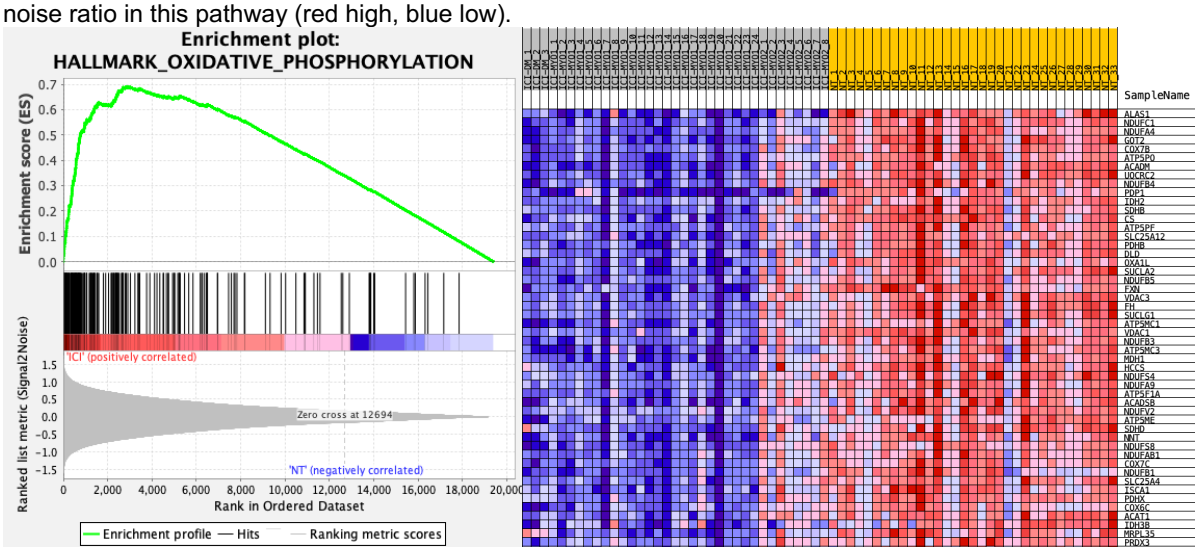
NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 15. Expression (log2[TMM+1]) of representative gene markers associated with muscle regeneration, adult skeletal muscle, oxidative phosphorylation, and mitochondrial function.

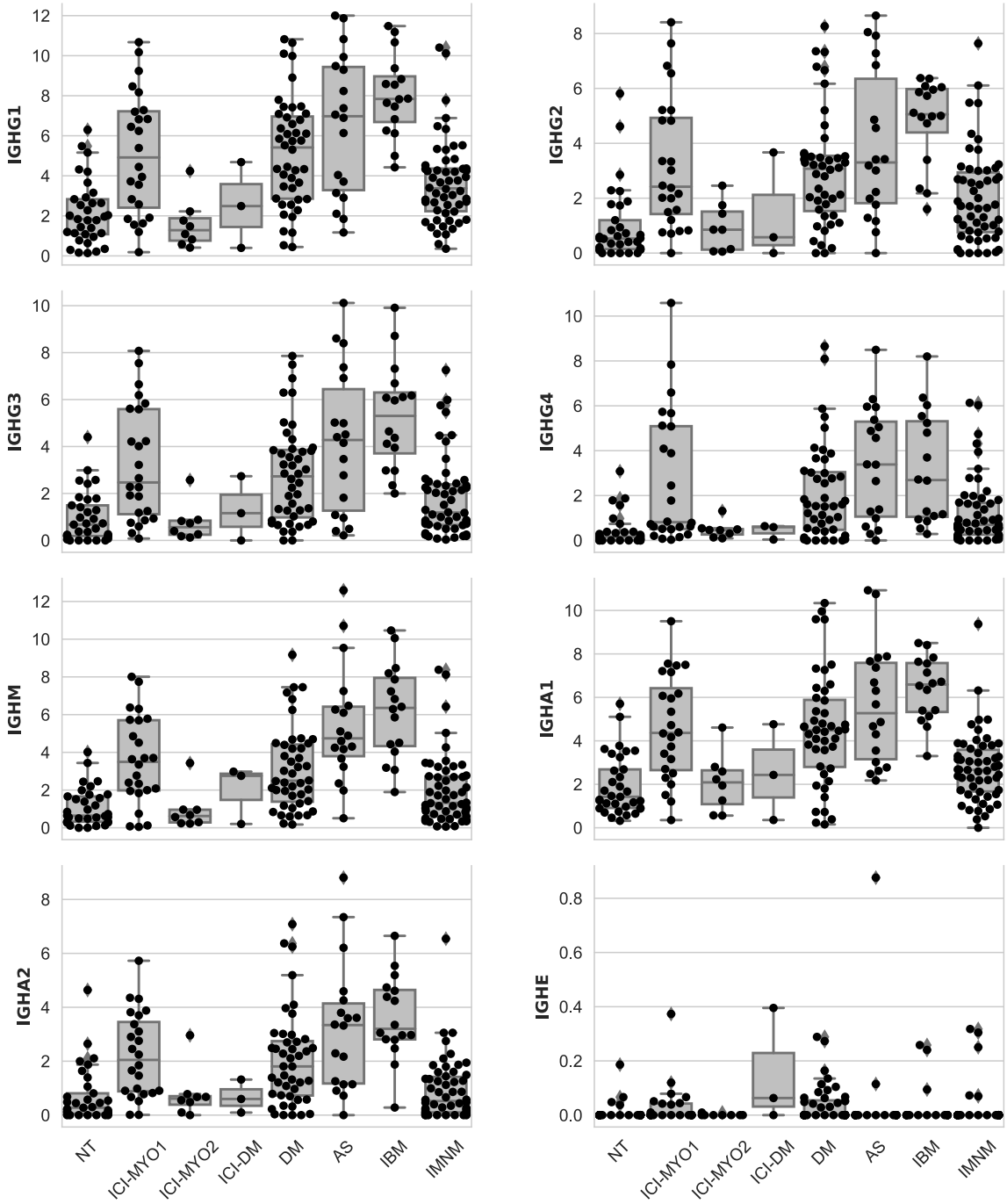


NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 16. Gene Set Enrichment Analysis of the oxidative phosporylation pathway (left) in immune checkpoint-induced myopathy patients compared to normal muscle (p-value 0.006). Fifty genes with the highest signal-to-noise ratio in this pathway (red high, blue low).



Supplementary Figure 17. Expression (log2[TMM+1]) of immunoglobulin genes.



NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 18. Expression (average z-score of log2[TMM+1]) of TNF receptors and their ligands showing general overexpression shared amongst different types of inflammatory myopathy.

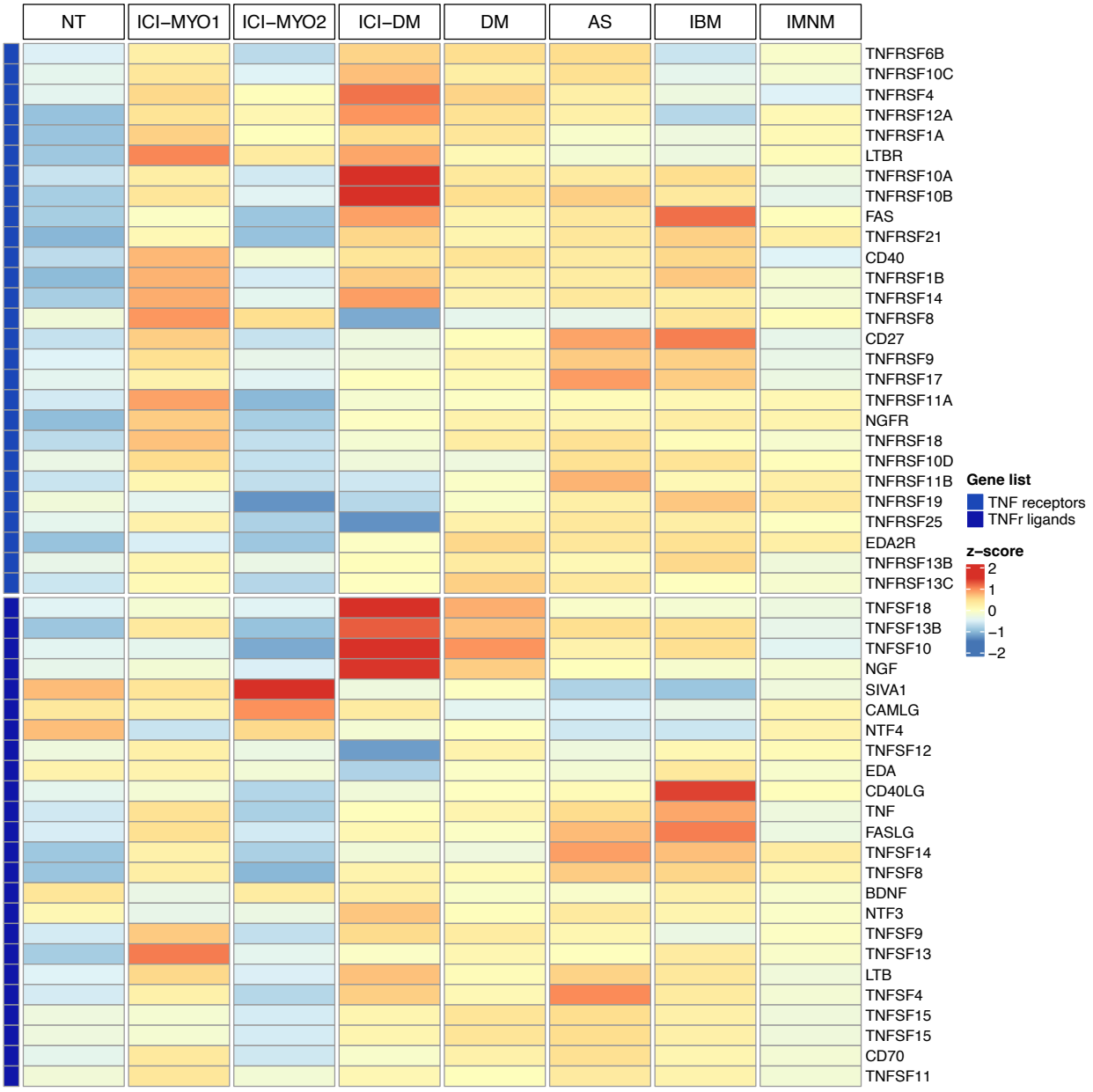
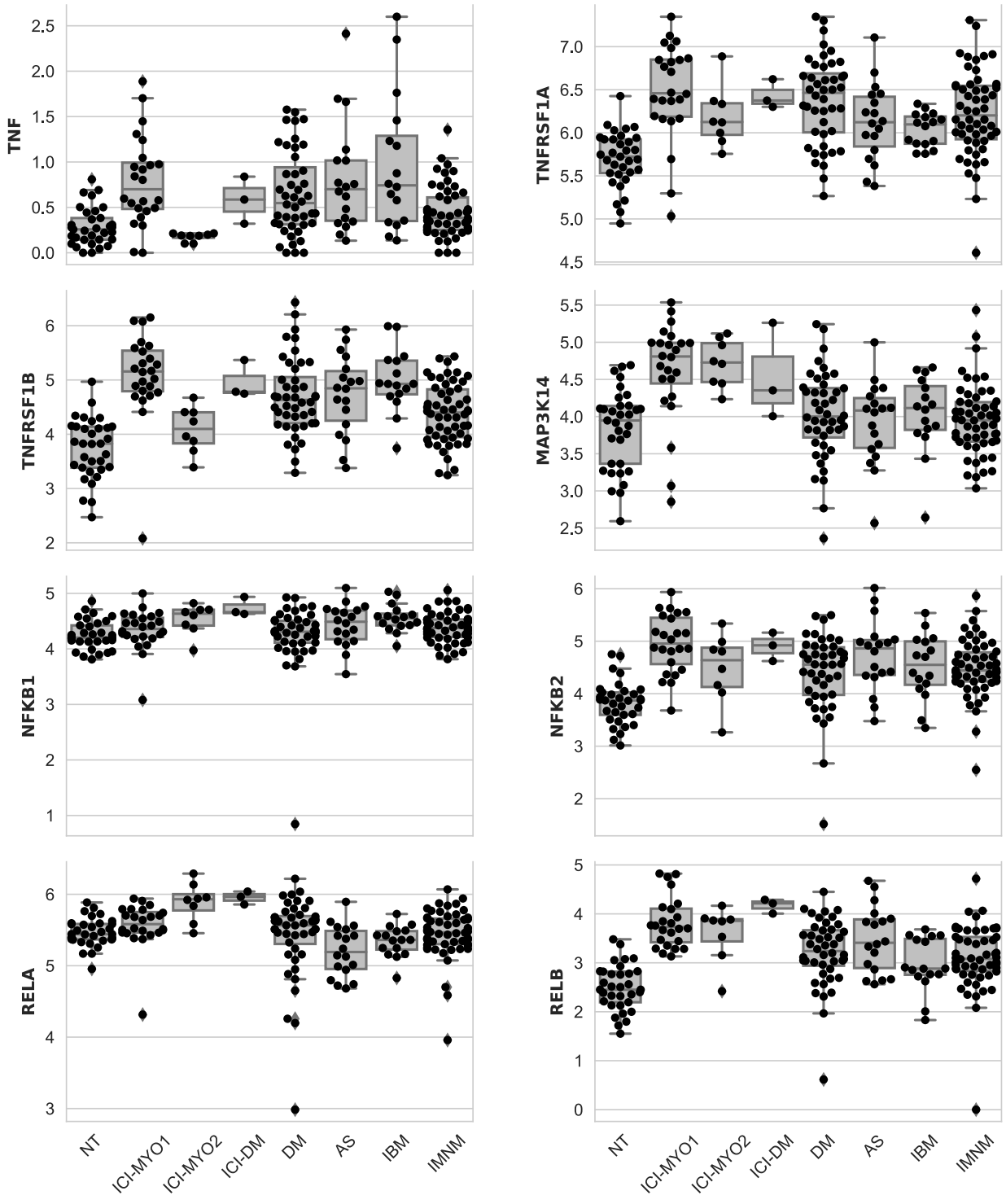
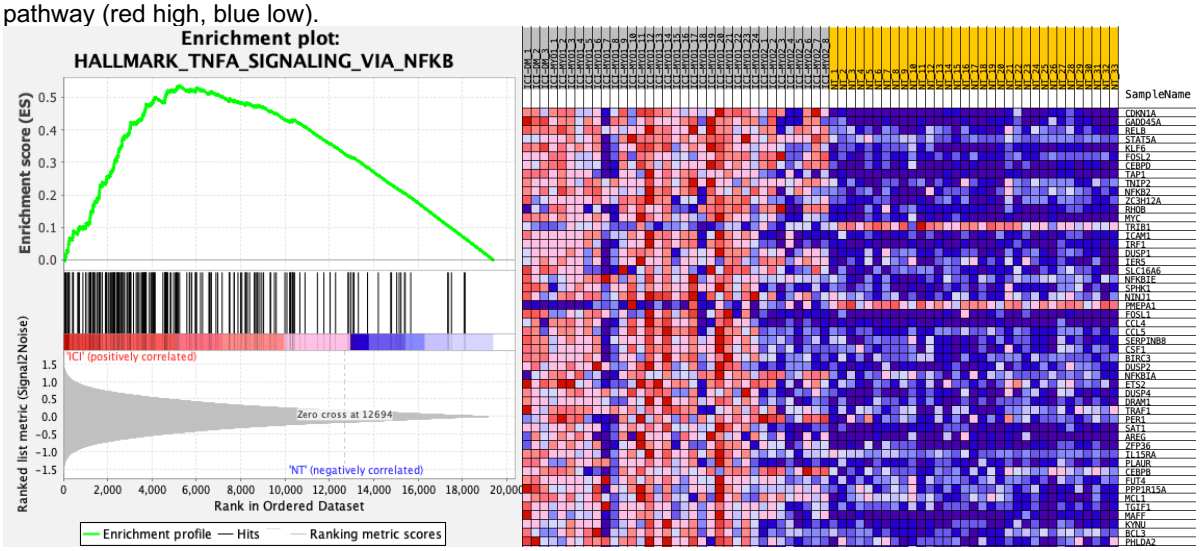


Figure 19. Expression (log2[TMM+1]) of representative genes from the TNF pathway.

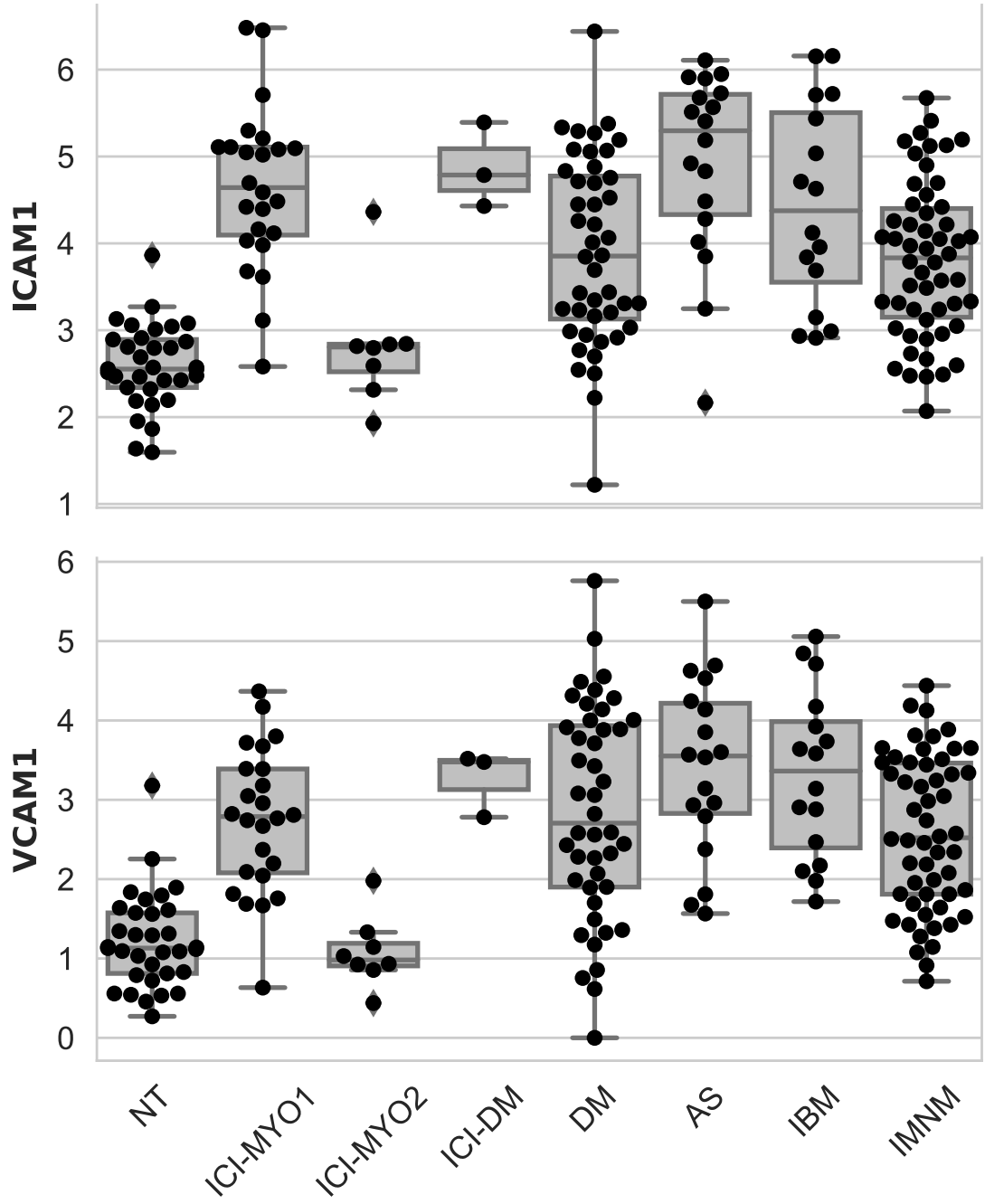


NT: normal muscle; ICI-PM: immune checkpoint-induced myopathy with no skin involvement; ICI-DM: immune checkpoint-induced dermatomyositis; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy.

Supplementary Figure 20. Gene Set Enrichment Analysis of the TNFA pathway (left) in immune checkpoint-induced myopathy patients compared to normal muscle (p-value 0.01). Fifty genes with the highest signal-to-noise ratio in this pathway (red high, blue low).

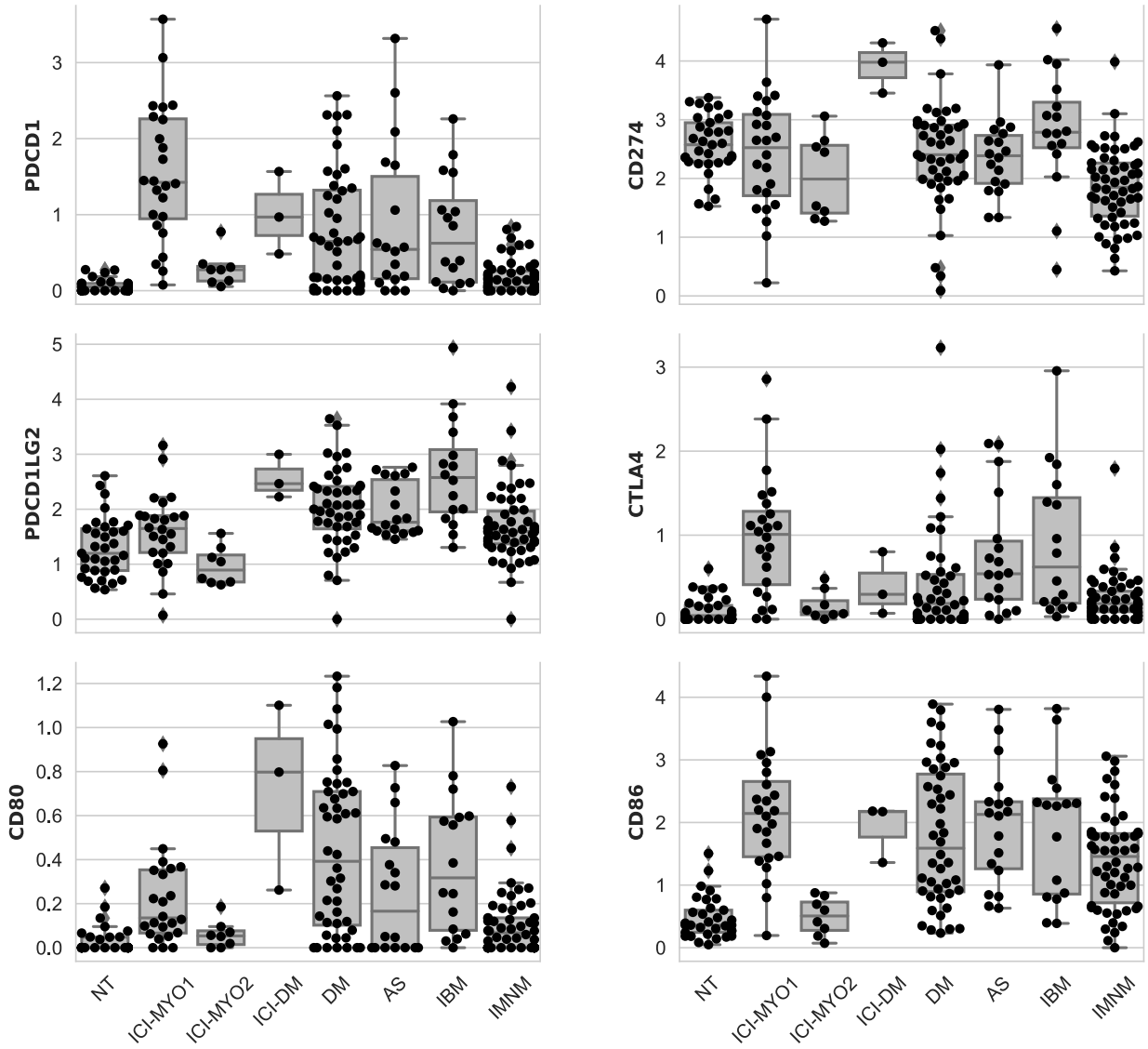


Supplementary Figure 21. Expression (log2[TMM+1]) of ICAM1 and VCAM1.



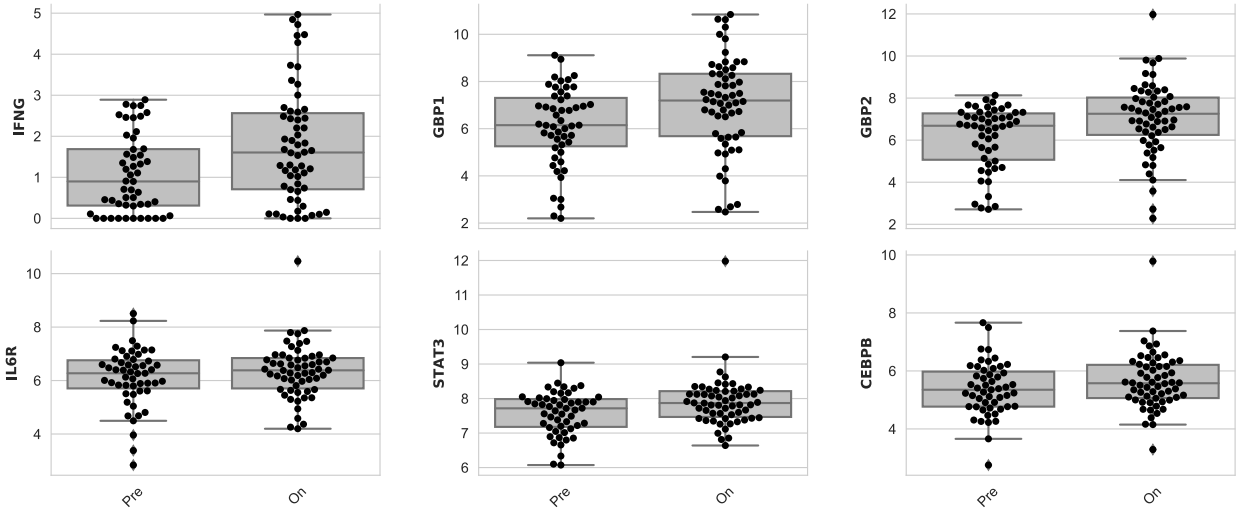
NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy

Supplementary Figure 22. Expression (log2[TMM+1]) of checkpoint genes.



NT: normal muscle; DM: dermatomyositis; AS: antisynthetase syndrome; IBM: inclusion body myositis; IMNM: immune-mediated necrotizing myopathy

Supplementary Figure 23. Expression levels (log2[TMM + 1]) of representative genes of the IFNG (row 1) and IL6 (row 2) pathways. Tumors treated with immune checkpoint inhibitors (ICI) have overexpression of IFNG and IFNG-stimulated genes (row 1) but not of genes related to the IL6 pathway (row 2).



Pre: pre-treatment; On: on-treatment.

Supplementary Table 1. Patient characteristics.

Group	Female	Age at biopsy	Dermatomyositis	Infl. infiltrates	Type of tumor	Treat. cycle	PD1 inhibitor	PD-L1 inhibitor	CTLA4 inhibitor	Myocarditis	Diplopia	Dysphagia	Anti-striational	Anti-AChR	Other atbs	CK at biopsy	Peak CK
ICI_DM	no	73	yes	yes	Urothelial carcinoma	2	yes	no	no	no	no	no	NA	NA	Anti-TIF1g	428	428
ICI_DM	no	44	yes	yes	Melanoma	1	yes	no	no	no	no	no	no	NA	Anti-TIF1g	9088	9088
ICI_DM	yes	75	yes	yes	Urothelial carcinoma	2	no	yes	no	no	no	no	no	NA	Anti-TIF1g	129	129
ICI_MYO1	no	63	no	yes	Pancreatic adenocarcinoma	1	no	yes	no	yes	yes	no	NA	yes	Anti-Ro60, anti-Ro52, anti-SRP, anti-Mi2a	2889	4204
ICI_MYO1	no	69	no	yes	Liposarcoma	1	yes	no	no	no	no	no	no	yes	Anti-MuSK	172	14000
ICI_MYO1	no	72	no	yes	Thyroid anaplastic	1	no	yes	yes	yes	yes	no	yes	no	Anti-Jo1	912	23495
ICI_MYO1	no	65	no	yes	Pancreatic neuroendocrine	2	no	yes	yes	no	yes	no	yes	yes	Anti-SRP	273	2430
ICI_MYO1	no	82	no	yes	Melanoma	1	yes	no	no	yes	no	no	yes	yes	NA	85	85
ICI_MYO1	no	60	no	yes	Squamous cell lung carcinoma	3	no	yes	no	no	no	no	yes	yes	None	201	201
ICI_MYO1	no	74	no	yes	Thymoma	4	no	yes	no	no	no	no	yes	no	Anti-PM/Scl100	597	1150
ICI_MYO1	no	63	no	yes	Squamous cell lung carcinoma	3	yes	no	no	no	yes	no	no	no	Anti-TIF1g, anti-Ro52	380	380
ICI_MYO1	yes	61	no	yes	Squamous head neck carcinoma	3	yes	no	no	no	no	no	no	no	Anti-HMGCR	1900	5976
ICI_MYO1	no	86	no	yes	Cholangiocarcinoma	2	no	yes	no	yes	no	no	yes	yes	Anti-TIF1g	638	11953
ICI_MYO1	no	67	no	yes	Lung adenocarcinoma	2	yes	no	no	yes	no	yes	yes	no	Anti-PM/Scl, anti-M2	10850	10850
ICI_MYO1	yes	62	no	yes	Breast adenocarcinoma	1	yes	no	no	yes	yes	no	yes	yes	None	11830	11830
ICI_MYO1	yes	60	no	yes	Melanoma	1	yes	no	no	no	no	no	no	NA	Anti-M2	1184	1184
ICI_MYO1	yes	66	no	yes	Lung adenocarcinoma	2	yes	no	no	yes	no	no	yes	yes	Anti-NPX2, anti-Ro52	1934	1934
ICI_MYO1	yes	58	no	yes	Melanoma	3	yes	no	no	no	no	no	NA	NA	None	666	666
ICI_MYO1	yes	81	no	yes	Melanoma	1	yes	no	no	yes	no	no	no	no	Anti-NTSc1A	8333	8333
ICI_MYO1	no	55	no	no	Melanoma	2	yes	no	no	yes	no	no	no	no	NA	280	1284
ICI_MYO1	no	57	no	yes	Melanoma	2	yes	no	no	no	no	no	yes	no	NA	180	444
ICI_MYO1	no	49	no	yes	Esophageal adenocarcinoma	2	yes	no	no	yes	yes	no	no	no	NA	180	333
ICI_MYO1	no	84	no	no	Merkel cell carcinoma	1	yes	no	no	no	no	yes	yes	yes	None	1226	2685
ICI_MYO1	no	78	no	no	Thymoma	1	no	yes	no	yes	no	no	NA	yes	None	1195	1195
ICI_MYO1	yes	68	no	yes	Thymoma	1	no	yes	no	yes	no	no	NA	yes	None	977	977
ICI_MYO1	yes	59	no	NA	Thymoma	2	no	yes	no	no	no	no	NA	no	None	1086	1086
ICI_MYO1	no	50	no	NA	Thymoma	1	no	yes	no	yes	no	no	NA	yes	None	880	1084
ICI_MYO2	no	76	no	yes	Melanoma	2	yes	no	no	no	no	no	NA	yes	Anti-Mi2b, anti-NXP2	1169	4275
ICI_MYO2	yes	65	no	no	Mesothelioma	3	yes	no	no	no	no	no	NA	NA	NA	51	481
ICI_MYO2	no	69	no	no	Pancreatic adenocarcinoma	1	yes	no	yes	no	no	no	no	no	NA	5599	6260
ICI_MYO2	no	69	no	no	Melanoma	1	yes	no	no	no	no	no	yes	no	NA	427	7307
ICI_MYO2	no	46	no	no	Follicular lymphoma	2	yes	no	no	no	yes	no	yes	yes	NA	72	72
ICI_MYO2	no	70	no	yes	Melanoma	1	yes	no	no	no	no	yes	yes	no	None	NA	542
ICI_MYO2	yes	88	no	no	Renal cell carcinoma	2	yes	no	yes	no	no	yes	no	no	None	26	28
ICI_MYO2	no	68	no	yes	Renal cell carcinoma	1	yes	no	yes	no	no	no	yes	no	NA	492	762

Supplementary Table 2. Type of immune checkpoint inhibitor.

Group	Nivolumab	Pembrolizumab	Atezolizumab	Avelumab	Durvalumab	Tremelimumab	Ipilimumab	M7824
ICI_DM	no	yes	no	no	no	no	no	no
ICI_DM	yes	no	no	no	no	no	no	no
ICI_DM	no	no	no	yes	no	no	no	no
ICI_MYO1	no	no	yes	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	no	no	no	yes	yes	no	no
ICI_MYO1	no	no	no	no	yes	yes	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	no	no	no	yes	no	no	no
ICI_MYO1	no	no	yes	no	no	no	no	no
ICI_MYO1	yes	no	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	no	no	no	no	no	no	yes
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	yes	no	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	yes	no	no	no	no	no	no	no
ICI_MYO1	yes	no	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	yes	no	no	no	no	no	no
ICI_MYO1	no	no	no	yes	no	no	no	no
ICI_MYO1	no	no	no	yes	no	no	no	no
ICI_MYO1	no	no	no	yes	no	no	no	no
ICI_MYO1	no	no	no	yes	no	no	no	no
ICI_MYO2	no	yes	no	no	no	no	no	no
ICI_MYO2	no	yes	no	no	no	no	no	no
ICI_MYO2	yes	no	no	no	no	no	yes	no
ICI_MYO2	no	yes	no	no	no	no	no	no
ICI_MYO2	no	yes	no	no	no	no	no	no
ICI_MYO2	yes	no	no	no	no	no	no	no
ICI_MYO2	yes	no	no	no	no	no	yes	no
ICI_MYO2	yes	no	no	no	no	no	yes	no

Supplementary Table 3. Differential expression of relevant genes related to immune-checkpoint inhibitor myopathy by bulk RNA sequencing. The different groups identified by the unsupervised clustering analysis (ICI-DM, ICI-MYO1, and ICI-MYO2) were compared to normal muscle biopsies (NT) and to each other. Missing values correspond to genes that did not pass the cutoff to be included in the differential expression.

	ICI-MYO1 vs. NT		ICI-MYO2 vs. NT		ICI-DM vs. NT		ICI-MYO1 vs. ICI-MYO2		ICI-DM vs. ICI-MYO1		ICI-DM vs. ICI-MYO2	
Gene	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**
ACTA1	-2.2	5e-08	-0.8	2e-05	-3.4	7e-16	-1.3	0.1	-1.2	0.6	-2.6	0.002
BDNF	-1.1	0.07	0.1	0.9	0.3	0.8	-1.3	0.07	1.5	0.3	0.1	0.9
CAMLG	-0.0	0.6	0.2	0.1	-0.0	1	-0.2	0.1	0.1	0.9	-0.2	0.3
CD14	2.3	1e-07	-0.7	0.05	1.9	2e-05	3.0	0.002	-0.5	0.8	2.5	9e-04
CD19		NA		NA		NA	2.5	0.007	-0.0	1		NA
CD27	2.8	1e-05		NA	1.4	0.03	3.2	0.008	-1.5	0.6	1.7	0.2
CD274	-0.3	0.4	-0.9	0.004	1.5	6e-06	0.5	0.6	1.9	0.1	2.3	0.003
CD3E	1.9	0.01	-0.4	0.5	1.8	0.001	2.2	0.2	-0.2	1	2.0	0.1
CD4	2.8	2e-11	-0.7	0.06	1.5	8e-04	3.5	2e-04	-1.2	0.3	2.2	0.01
CD40	0.9	8e-08	0.3	0.03	0.7	4e-04	0.6	0.04	-0.2	0.8	0.4	0.2
CD40LG		NA		NA	0.4	0.7	2.7	0.02	0.0	1		NA
CD68	3.1	1e-08	0.5	0.2	3.7	1e-09	2.5	0.01	0.6	0.8	3.1	0.001
CD70		NA		NA	1.5	0.06	3.0	0.002	-0.5	0.8		NA
CD80		NA		NA		NA	2.2	0.02	2.2	0.04	4.4	0.002
CD86	3.1	1e-06	-0.2	0.7	3.1	5e-10	3.2	0.007	0.1	1	3.2	0.002
CD8A	3.5	8e-06	0.5	0.3	2.4	6e-05	3.0	0.03	-1.2	0.7	1.7	0.2
CEBPB	0.8	5e-04	1.4	1e-04	1.0	0.05	-0.6	0.2	0.2	0.9	-0.5	0.4
COX7B	-1.2	1e-14	-0.7	2e-04	-1.7	7e-09	-0.6	0.04	-0.4	0.5	-1.1	0.02
COX7C	-0.9	1e-12	-0.3	0.05	-1.2	7e-06	-0.6	0.006	-0.3	0.6	-0.9	0.006
CTLA4		NA		NA		NA	3.2	0.04	-1.2	0.7	2.0	0.2
EDA	0.2	0.5	-0.4	0.05	-0.7	0.007	0.5	0.3	-0.8	0.4	-0.4	0.5
EDA2R	0.9	0.005	0.1	0.9	1.5	0.001	0.8	0.2	0.7	0.5	1.5	0.1
EGR1	2.8	2e-05	0.1	0.9	3.2	0.01	2.7	0.01	0.2	1	3.0	0.03
FAS	0.8	2e-04	-0.2	0.5	1.5	2e-06	0.9	0.02	0.8	0.2	1.7	0.003
FASLG		NA		NA	2.7	6e-05	2.5	0.05	0.1	1	2.7	0.006
FOS	3.5	9e-06	0.7	0.5	3.6	0.02	2.9	0.01	-0.0	1	2.8	0.05
FOSB	4.6	4e-06	1.0	0.2	3.5	5e-04	3.8	0.03	-1.3	0.7	2.6	0.05
FOSL1	4.2	2e-08	1.4	0.006	5.8	7e-15	2.7	0.005	1.7	0.1	4.3	4e-07
GBP2	2.5	1e-13	1.2	6e-06	3.0	1e-11	1.3	0.04	0.5	0.8	1.8	0.01
GZMA	2.3	0.004	-0.3	0.5	1.9	4e-04	2.4	0.1	-0.5	0.9	2.0	0.08
GZMB	2.7	0.009	0.4	0.5	3.7	3e-09	2.2	0.2	1.0	0.8	3.1	0.004
ICAM1	2.3	4e-13	0.1	0.8	2.5	1e-10	2.2	2e-04	0.2	0.9	2.4	7e-04
IFI30	5.0	4e-13	0.4	0.7	5.2	2e-10	4.6	9e-04	0.2	0.9	4.7	0.02
IFNA1		NA		NA		NA		NA		NA		NA
IFNA10		NA		NA		NA		NA		NA		NA
IFNA13		NA		NA		NA		NA		NA		NA
IFNA14		NA		NA		NA		NA		NA		NA
IFNA16		NA		NA		NA		NA		NA		NA
IFNA17		NA		NA		NA		NA		NA		NA
IFNA2		NA		NA		NA		NA		NA		NA
IFNA21		NA		NA		NA		NA		NA		NA
IFNA4		NA		NA		NA		NA		NA		NA
IFNA5		NA		NA		NA		NA		NA		NA
IFNA6		NA		NA		NA		NA		NA		NA
IFNA7		NA		NA		NA		NA		NA		NA
IFNA8		NA		NA		NA		NA		NA		NA
IFNAR1	-0.5	7e-06	-0.4	0.003	-0.4	0.1	-0.1	0.7	0.2	0.7	0.0	0.9
IFNAR2	0.4	4e-04	-0.5	0.002	-1.5	0.5	0.9	8e-04	-1.4	0.3	0.5	0.09
IFNB1		NA		NA		NA		NA		NA		NA
IFNG		NA		NA		NA	3.5	0.03	-1.7	0.7	1.9	0.3
IGHA1	3.0	2e-04	-0.3	0.7	0.9	0.5	3.2	0.05	-2.1	0.6	1.0	0.6
IGHA2	2.5	0.05	0.1	1	1.2	0.4	2.3	0.3	-1.5	0.8	0.9	0.7
IGHE		NA		NA		NA		NA		NA		NA
IGHG1	3.4	6e-04	-1.2	0.2	0.8	0.6	4.4	0.04	-2.6	0.6	1.7	0.3
IGHG2	3.0	0.03	-0.6	0.6	0.8	0.7	3.2	0.2	-2.6	0.8	0.8	0.8
IGHG3	3.3	0.008	-0.3	0.8	0.9	0.6	3.4	0.1	-2.8	0.7	0.8	0.7
IGHG4	3.9	0.02	0.7	0.3	1.1	0.4	2.9	0.3	-2.9	0.7	0.1	0.9
IGHM	2.9	0.01	-0.8	0.3	1.7	0.08	3.4	0.1	-1.1	0.9	2.2	0.2
IL17RC	0.5	0.01	0.8	0.02	-0.1	0.9	-0.3	0.2	-0.6	0.3	-0.9	0.05
IL6	4.4	4e-05		NA	5.2	1e-10	3.4	0.04	0.8	0.8	4.2	0.002
IL6R	1.3	4e-07	1.9	2e-07	0.9	0.02	-0.7	0.2	-0.3	0.8	-1.1	0.2
ISG15	2.1	5e-09	0.4	0.2	9.0	3e-22	1.6	0.006	6.9	2e-05	8.5	1e-08
JUN	0.5	0.04	0.4	0.1	0.7	0.07	0.1	0.8	0.2	0.9	0.3	0.5
JUNB	3.1	1e-08	0.7	0.3	3.1	8e-04	2.5	0.007	-0.1	1	2.3	0.01
LTB	2.4	4e-06	0.2	0.7	2.9	6e-05	2.1	0.009	0.5	0.8	2.5	0.01

	ICI-MYO1 vs. NT		ICI-MYO2 vs. NT		ICI-DM vs. NT		ICI-MYO1 vs. ICI-MYO2		ICI-DM vs. ICI-MYO1		ICI-DM vs. ICI-MYO2	
Gene	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**
LTBR	1.2	5e-10	0.7	4e-05	1.1	3e-05	0.5	0.2	-0.1	0.9	0.3	0.2
MAP3K14	0.9	1e-05	1.0	2e-04	0.8	0.04	-0.1	0.8	-0.0	1	-0.2	0.6
MS4A1		NA		NA	2.1	0.02	2.6	0.07	0.9	0.8	3.5	0.04
MT-TI	-2.2	1e-08	-1.5	0.001	-2.7	0.005	-0.7	0.1	-0.4	0.8	-1.1	0.1
MT-TL1	-2.9	5e-17	-2.2	1e-07	-2.8	6e-06	-0.8	0.04	0.2	0.9	-0.6	0.3
MX1	0.7	0.007	-0.6	0.03	7.0	2e-19	1.3	0.02	6.3	2e-05	7.6	4e-08
MYH1	-1.0	0.09	0.3	0.7	-2.9	0.008	-1.3	0.3	-1.8	0.4	-3.2	0.02
MYH3	2.7	5e-07	1.0	0.07	0.8	0.3	1.8	0.07	-1.9	0.3	-0.3	0.8
NCAM1	1.5	3e-06	-0.0	0.9	1.1	0.02	1.5	0.01	-0.4	0.8	1.1	0.05
NFKB1	0.1	0.3	0.3	0.03	0.5	0.003	-0.2	0.4	0.5	0.3	0.2	0.4
NFKB2	1.2	4e-11	0.7	0.004	1.2	1e-04	0.5	0.09	-0.0	1	0.4	0.4
NGF	0.1	0.7	-0.4	0.4	1.9	2e-05	0.5	0.3	1.8	0.02	2.3	0.004
NGFR	2.9	2e-09	0.2	0.8	2.0	0.004	2.7	8e-04	-0.9	0.5	1.7	0.09
NTF3	-0.4	0.1	-0.8	0.04	0.3	0.6	0.2	0.6	0.8	0.4	1.0	0.3
NTF4	-1.4	4e-05	-0.2	0.4	-0.7	0.03	-1.2	0.06	0.8	0.7	-0.5	0.4
PDCD1	5.1	4e-10		NA	4.6	2e-12	3.1	0.006	-0.6	0.8	2.5	0.01
PDCD1LG2	0.3	0.6	-0.9	0.01	1.7	1e-05	1.0	0.2	1.5	0.3	2.5	4e-04
PRF1	2.1	2e-04	0.4	0.3	2.8	1e-06	1.6	0.1	0.7	0.8	2.3	0.008
RELA	0.1	0.1	0.4	7e-04	0.5	0.001	-0.3	0.1	0.4	0.2	0.1	0.8
RELB	1.5	2e-13	1.3	2e-06	1.9	1e-08	0.3	0.4	0.4	0.5	0.6	0.2
SIVA1	-0.2	0.2	0.5	0.02	-0.6	0.05	-0.7	0.02	-0.4	0.4	-1.1	0.01
SOCS3	4.4	2e-11	1.2	0.02	4.5	4e-08	3.3	0.001	-0.0	1	3.3	0.002
STAT3	0.4	4e-05	0.5	5e-04	0.8	5e-04	-0.2	0.4	0.4	0.3	0.2	0.5
TGFB1	1.6	2e-10	-0.1	0.6	0.8	0.01	1.7	4e-04	-0.7	0.4	1.0	0.03
TGFB2	-0.5	0.1	-1.0	0.002	1.2	0.002	0.5	0.4	1.7	0.06	2.2	0.005
TGFB3	0.1	0.8	-0.0	1	1.1	1e-04	0.0	1	1.1	0.3	1.1	0.02
TGFBR2	0.9	1e-04	0.1	0.7	0.8	0.06	0.8	0.07	-0.1	0.9	0.6	0.06
TNF	1.5	0.01	-0.5	0.3	1.6	0.008	2.0	0.07	0.0	1	2.0	0.004
TNFRSF10A	1.3	1e-04	0.2	0.6	2.7	2e-10	1.1	0.04	1.4	0.07	2.5	4e-05
TNFRSF10B	1.0	2e-07	0.3	0.2	1.9	1e-09	0.7	0.03	0.9	0.08	1.6	0.002
TNFRSF10C	2.0	3e-05		NA	2.6	2e-04	1.6	0.02	0.5	0.7	2.2	0.02
TNFRSF10D	0.8	0.003	-0.4	0.1	0.1	0.8	1.2	0.03	-0.7	0.6	0.4	0.4
TNFRSF11A	2.0	6e-06	-1.3	0.005	0.7	0.1	3.2	9e-04	-1.2	0.5	2.0	0.02
TNFRSF11B	1.7	2e-04	0.3	0.6	0.3	0.8	1.4	0.02	-1.4	0.3	-0.0	1
TNFRSF12A	2.1	2e-08	1.7	0.001	2.9	1e-04	0.4	0.4	0.7	0.5	1.2	0.06
TNFRSF13B		NA		NA		NA		NA		NA		NA
TNFRSF13C	1.4	2e-08	-0.3	0.4	1.1	0.006	1.8	2e-04	-0.2	0.8	1.5	0.02
TNFRSF14	1.5	5e-07	0.4	0.2	1.6	1e-04	1.1	0.05	0.1	1	1.2	0.08
TNFRSF17		NA		NA		NA	1.6	0.2	0.8	0.8		NA
TNFRSF18	3.1	7e-10		NA	2.2	5e-04	2.8	6e-04	-1.0	0.5	1.8	0.04
TNFRSF19	-0.0	0.9	-1.6	5e-05	-0.6	0.1	1.5	0.008	-0.5	0.6	0.9	0.4
TNFRSF1A	0.8	1e-08	0.5	0.004	0.8	0.001	0.3	0.2	-0.0	1	0.3	0.4
TNFRSF1B	1.5	1e-09	0.3	0.2	1.3	4e-04	1.1	0.008	-0.1	0.9	0.9	0.02
TNFRSF21	1.1	2e-07	0.1	0.8	1.4	2e-06	1.0	0.005	0.4	0.6	1.3	0.006
TNFRSF25	0.6	6e-04	-0.3	0.2	-0.7	0.1	0.9	0.003	-1.2	0.03	-0.4	0.5
TNFRSF4	1.2	0.001	0.8	0.06	2.2	6e-06	0.4	0.6	1.0	0.4	1.4	0.1
TNFRSF6B	0.7	0.3	-2.1	0.1	1.8	0.02	2.9	0.2	1.1	0.7	3.9	0.1
TNFRSF8	1.2	0.001	0.7	0.03	-1.5	0.01	0.4	0.5	-2.6	0.1	-2.2	0.05
TNFRSF9		NA		NA		NA	1.7	0.04	-1.3	0.5	0.4	0.7
TNFSF10	-0.0	1	-0.8	6e-04	1.8	6e-08	0.8	0.2	1.9	0.1	2.6	1e-04
TNFSF11		NA		NA		NA		NA		NA		NA
TNFSF12	0.3	0.07	-0.1	0.7	-0.5	0.06	0.3	0.3	-0.8	0.3	-0.5	0.3
TNFSF13	2.1	7e-11	0.6	0.04	1.0	0.01	1.5	0.002	-1.1	0.3	0.4	0.2
TNFSF13B	2.3	3e-05	-0.4	0.3	3.8	2e-14	2.5	0.02	1.6	0.3	4.1	2e-05
TNFSF14	2.4	1e-05	0.3	0.6	2.0	0.002	2.0	0.02	-0.4	0.9	1.6	0.04
TNFSF15	0.2	0.4	-0.7	0.1	0.7	0.3	0.9	0.03	0.5	0.6	1.3	0.1
TNFSF18		NA		NA	5.3	1e-15		NA	4.0	2e-05	5.7	7e-05
TNFSF4	1.3	2e-05	-0.3	0.5	1.8	3e-04	1.6	0.002	0.5	0.6	2.0	0.01
TNFSF8	1.9	0.01	-0.5	0.4	2.4	1e-05	2.3	0.08	0.5	0.9	2.8	0.002
TNFSF9	1.8	4e-07	-0.2	0.6	1.8	2e-04	2.0	0.001	-0.0	1	2.0	0.005
TYK2	0.3	0.01	0.5	0.002	-0.1	0.8	-0.2	0.3	-0.3	0.6	-0.6	0.04
VCAM1	2.2	2e-07	-0.4	0.3	2.8	5e-09	2.5	0.003	0.7	0.6	3.2	1e-04

Supplementary Table 4. Differential expression of relevant genes related to immune-checkpoint inhibitor myopathy by single-nuclei RNA sequencing. Four biopsies from the cluster ICI-MYO1 were compared with a representative selection of patients with other types of inflammatory myopathy (4 dermatomyositis, 3 antisynthetase syndrome, 6 immune-mediated necrotizing myositis, and 2 inclusion body myositis). Missing values correspond to genes that did not pass the cutoff to be included in the differential expression.

	T-cells ICI-MYO1 vs. other myositis		FAP cells ICI-MYO1 vs. other myositis		Adipocytes ICI-MYO1 vs. other myositis		Endothelial cells ICI-MYO1 vs. other myositis		Myeloid cells ICI-MYO1 vs. other myositis		Fibroblasts ICI-MYO1 vs. other myositis		Muscle fibers ICI-MYO1 vs. other myositis		Satellite cells ICI-MYO1 vs. other myositis	
Gene	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**	**avg_log2FC**	**p_val_adj**
CEBPB		NA		NA	0.3	0.1		NA		NA		NA	0.3	6e-35	0.6	2e-07
EGR1	0.3	0.7	1.0	5e-13	0.7	7e-32	0.6	4e-20	0.9	3e-25	1.0	2e-94	0.7	3e-263	1.2	9e-16
FOS	1.2	6e-15	1.1	3e-31	0.7	1e-32	0.7	6e-29	1.2	3e-14	1.3	4e-161	0.9	5e-280	1.5	8e-22
FOSB	0.5	3e-04	1.1	8e-20	1.5	6e-69	0.6	4e-28	1.8	6e-45	1.7	4e-191	0.4	1e-164	1.4	2e-38
GBP1		NA		NA		NA		NA		NA		NA		NA		NA
GBP2	0.4	1		NA		NA		NA		NA		NA		NA		NA
IFI30		NA		NA		NA		NA	0.6	1		NA		NA		NA
IFIT1		NA	0.4	1		NA	0.3	1		NA	0.3	0.9		NA		NA
IFNB1		NA		NA		NA		NA		NA		NA		NA		NA
IFNG		NA		NA		NA		NA		NA		NA		NA		NA
IL6		NA		NA		NA		NA		NA		NA		NA		NA
IL6R		NA		NA		NA		NA		NA		NA	0.3	2e-07		NA
ISG15		NA	0.5	0.08		NA	0.3	1	0.3	5e-04		NA		NA	0.3	1
JUN		NA	0.7	0.01	0.6	8e-16	0.5	1e-06	0.4	1	1.0	4e-55	0.3	1e-17	0.9	2e-05
JUNB		NA	0.5	1e-06	0.4	2e-11		NA		NA	0.3	6e-15	0.3	2e-53	0.7	7e-07
MX1		NA	0.4	1	0.6	1e-10	0.3	1		NA	0.6	5e-06		NA		NA
MX2		NA		NA	0.3	1e-06		NA	-0.3	1		NA		NA		NA
PSMB8		NA		NA		NA		NA		NA		NA		NA		NA
SOCS3		NA		NA	0.6	6e-31	0.4	2e-09		NA	0.5	2e-47		NA	0.3	1
STAT3		NA	0.3	1	0.7	2e-08		NA		NA		NA		NA		NA
TYK2		NA		NA		NA		NA		NA		NA		NA		NA

Supplementary Table 5. Differential expression of relevant genes related to immune-checkpoint inhibitor myopathy by single-nuclei RNA sequencing. Four biopsies from the cluster ICI-DM were compared with a representative selection of patients with other types of inflammatory myopathy (4 dermatomyositis, 3 antisynthetase syndrome, 6 immune-mediated necrotizing myositis, and 2 inclusion body myositis). Missing values correspond to genes that did not pass the cutoff to be included in the differential expression.

	T-cells ICI-DM vs. other myositis		FAP cells ICI-DM vs. other myositis		Adipocytes ICI-DM vs. other myositis		Endothelial cells ICI-DM vs. other myositis		Myeloid cells ICI-DM vs. other myositis		Fibroblasts ICI-DM vs. other myositis		Muscle fibers ICI-DM vs. other myositis		Satellite cells ICI-DM vs. other myositis	
Gene	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj	avg_log2FC	p_val_adj
CEBPB		NA		NA		NA	0.4	2e-17		NA		NA	0.3	6e-64		NA
EGR1		NA	1.2	9e-29	0.6	1e-26	0.5	7e-22	0.4	5e-16	1.1	3e-130	1.1	0	1.2	1e-21
FOS	0.7	5e-12	1.2	3e-32	0.8	1e-32	0.7	3e-36	1.1	8e-49	1.3	3e-200	1.3	0	1.6	1e-25
FOSB	0.4	1e-04	1.0	2e-16	1.7	3e-78	0.3	4e-15	0.9	3e-39	1.5	4e-180	0.3	3e-135	1.2	3e-17
GBP1		NA	0.6	2e-09	0.3	0.001	0.3	0.05	0.5	2e-12	0.8	9e-42		NA		NA
GBP2		NA	0.4	1		NA		NA	0.3	0.1		NA		NA		NA
IFI30		NA		NA		NA		NA	0.5	0.8		NA		NA		NA
IFIT1	1.0	2e-25	1.1	4e-34	0.4	5e-04	1.2	9e-32	1.0	3e-49	0.9	1e-76	1.1	0	1.4	1e-33
IFNB1		NA		NA		NA		NA		NA		NA		NA		NA
IFNG		NA		NA		NA		NA		NA		NA		NA		NA
IL6		NA		NA		NA		NA		NA		NA		NA		NA
IL6R		NA		NA		NA		NA		NA		NA		NA		NA
ISG15	2.2	2e-57	2.0	6e-53	0.8	4e-28	2.0	4e-79	2.1	5e-119	1.8	5e-210	2.2	0	2.2	2e-47
JUN		NA	1.1	7e-15	0.5	2e-12	0.4	1e-09	0.4	1e-04	1.1	9e-62	0.5	4e-106	0.6	0.05
JUNB		NA	0.6	2e-08	0.4	1e-08		NA	0.3	0.06	0.5	2e-43	0.5	7e-186	0.6	2e-06
MX1	1.8	8e-36	1.6	2e-43	0.5	5e-07	1.0	3e-17	2.0	1e-73	1.4	6e-89	1.6	0	1.8	4e-40
MX2	1.2	4e-17	1.2	1e-20		NA	1.0	7e-19	1.2	2e-36	1.2	1e-42	0.3	4e-106	1.2	6e-15
PSMB8		NA		NA		NA		NA		NA		NA		NA		NA
SOCS3		NA	0.6	5e-14	0.6	1e-29		NA		NA	0.6	5e-56		NA	0.4	1
STAT3		NA	0.7	0.04	0.6	0.5		NA		NA	0.3	1	0.4	8e-39		NA
TYK2		NA		NA		NA		NA		NA		NA		NA		NA

2.2. Complemento en miositis

Las proteínas del complemento se depositan en los músculos de los pacientes con miositis. Sin embargo, la expresión local y la regulación de los genes del complemento dentro del músculo de la miositis no están bien caracterizados todavía. En este estudio, análisis de secuenciación de RNA de muestras de biopsia muscular revelaron que los genes del complemento se sobreexpresan localmente y se correlacionan con marcadores de actividad de la enfermedad, incluida la expresión de genes inducidos por interferón-gamma. Análisis de secuenciación de RNA de células y núcleo único mostraron que la mayor parte de la expresión local de los genes del complemento se produce en macrófagos, fibroblastos y células satélite, y cada tipo de célula expresa diferentes conjuntos de genes del complemento. Las biopsias de pacientes con miopatía necrosante inmunomediada, que tienen los niveles más bajos de genes inducidos por IFNb, también tenían los niveles más bajos de expresión génica del complemento. Además, los estudios en células humanas en cultivo mostraron que el IFNg aumenta la expresión del complemento en macrófagos, fibroblastos y células musculares. En conjunto, nuestros resultados sugieren que en la miositis, el IFNg coordina la sobreexpresión local de los genes del complemento que se produce en varios tipos de células.

Coordinated local RNA overexpression of complement induced by interferon gamma in myositis

Maria Casal-Dominguez[a,b*], Iago Pinal-Fernandez[a,b*], Katherine Pak[a], Sandra Muñoz-Braceras[a], Jose C Milisenda[a,c,d], Jiram Torres-Ruiz[a], Stefania Del Orso[a], Faiza Naz[a], Gustavo Gutierrez-Cruz[a], Yaiza Duque-Jaimez[c], Ana Matas-Garcia[c,d,e], Laura Valls-Roca[c,d,e], Gloria Garrabou[c,d,e], Ernesto Trallero-Araguas[f,g], Brian Walitt[h], Lisa Christopher-Stine[b,i], Thomas E. Lloyd[b], Julie J. Paik[i], Jemima Albayda[i], Andrea Corse[b], Josep Maria Grau[c,d,e], Albert Selva-O'Callaghan[f,g], and Andrew L. Mammen[a,b,i]

**These authors contributed equally to this project.*

[a]Muscle Disease Unit, National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, Bethesda, MD, USA

[b]Department of Neurology, Johns Hopkins University School of Medicine, Baltimore, MD, USA

[c]Muscle Research Unit, Internal Medicine Service, Hospital Clinic, Barcelona, Spain

[d]Barcelona University, Barcelona, Spain

[e]CIBERER, Barcelona, Spain

[f]Systemic Autoimmune Disease Unit, Vall d'Hebron Institute of Research, Barcelona,

Spain

[g]Autonomous University of Barcelona, Barcelona, Spain

[h]National Institute of Neurological Disorders and Stroke, National Institutes of Health, Bethesda, MD, USA

[i]Department of Medicine, Johns Hopkins University School of Medicine, Baltimore, MD, USA

Address correspondence to: Andrew L. Mammen, M.D., Ph.D., or Iago Pinal-Fernandez, M.D., Ph.D., Ph.D., Muscle Disease Unit, Laboratory of Muscle Stem Cells and Gene Regulation, National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, 50 South Drive, Room 1141, Building 50, MSC 8024, Bethesda, MD 20892. E-mail: andrew.mammen@nih.gov or iago.pinalfernandez@nih.gov. Phone: 301-451-1199. Fax: 301-594-0305.

Acknowledgments: This study was funded, in part, by the Intramural Research Program of the National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health.

Keywords: Myositis, complement, RNA-sequencing, muscle biopsy

SUMMARY

Complement proteins are deposited in the muscles of patients with myositis. However, the local expression and regulation of complement genes within myositis muscle have not been well characterized. In this study, bulk RNA sequencing (RNAseq) analyses of muscle biopsy specimens revealed that complement genes are locally overexpressed and correlate with markers of myositis disease activity, including the expression of interferon-gamma (IFNγ)-induced genes. Single cell and single nuclei RNAseq analyses showed that most local expression of complement genes occurs in macrophages, fibroblasts, and satellite cells, with each cell type expressing different sets of complement genes. Biopsies from immune-mediated necrotizing myopathy patients, who have the lowest levels of IFNγ-induced genes, also had the lowest complement gene expression levels. Furthermore, data from cultured human cells showed that IFNγ upregulates complement expression in macrophages, fibroblasts, and muscle cells. Taken together, our results suggest that in myositis muscle, IFNγ coordinates the local overexpression of complement genes that occurs in several cell types.

INTRODUCTION

The complement system is a set of soluble proteins that are part of the innate immune defense, connecting innate and adaptive immunity. There are nine central complement components (C1-C9) as well as numerous complement activators and regulators[1]. Complement proteins are mainly produced in the liver. However, many other cell types can produce complement proteins, in some cases after stimulation by cytokines[2]. Importantly, dysregulated activation of the complement system may contribute to autoimmune diseases like type II glomerulonephritis, age-related macular degeneration, and atypical hemolytic uremic syndrome[1,3].

Complement deposition has been proposed to play a key pathogenic role in several neuromuscular diseases including the inflammatory myopathies (IM), a heterogenous family of muscle diseases that includes dermatomyositis (DM), the antisynthetase syndrome (AS), immune-mediated necrotizing myopathy (IMNM)[4–6], and inclusion body myositis (IBM). While each type of IM has its own characteristic clinical, serological, and muscle biopsy features, the deposition of complement proteins has been described in muscle tissues from each. For example, muscle biopsies from DM patients reveal C3b, C4b, and C5b-9 (the membrane attack complex; MAC) on endomysial capillaries[7,8]. In contrast, IMNM biopsies include MAC deposition on necrotic muscle fibers, atrophic muscle fibers, small arteries, veins, and capillaries within the muscle tissue[6,9–12].

Although a recent study using bulk RNAseq data showed that C1QB and C1QC genes are overexpressed in DM muscle tissue[13], it remains unclear which cells within DM muscle tissue contribute to complement gene overexpression. Indeed, a comprehensive analysis of complement gene expression within each type of IM muscle tissue has not been described. In the current study, our objectives were to a) quantify the local expression of complement genes in different types of IM, b) study the association between complement expression and IM disease activity, c) identify the types of cells expressing each comple-

ment gene in IM muscle, and d) determine how local complement gene expression might be regulated within IM muscle tissue.

METHODS

Patients

In this study, we included all muscle biopsies from patients enrolled in institutional review board-approved (IRB) longitudinal cohorts from the National Institutes of Health in Bethesda, MD; the Johns Hopkins Myositis Center in Baltimore, MD; the Vall d'Hebron Hospital, and the Clinic Hospital in Barcelona if they fulfilled Lloyd's criteria for IBM[14], or they fulfilled the Casal and Pinal criteria for other types of IM[15], and were positive for one of the following myositis specific autoantibodies (MSA): anti-Jo1, anti-NXP2, anti-Mi2, anti-TIF1g, anti-MDA5, anti-SRP or anti-HMGCR. Autoantibody testing was performed using one or more of the following techniques: ELISA, immunoprecipitation of proteins generated by *in vitro* transcription and translation (IVTT-IP), line blotting (EUROLINE myositis profile), or immunoprecipitation from ^{35}S-methionine-labeled HeLa cell lysates. We classified the patients as antisynthetase syndrome if they were positive for anti-Jo1 autoantibodies, as DM if they tested positive for anti-Mi2, anti-NXP2, anti-MDA5, or anti-TIF1g, and as IMNM if they had autoantibodies against SRP or HMGCR. We obtained histologically normal muscle biopsies to use as healthy comparators from the Johns Hopkins Neuromuscular Pathology Laboratory (n=12), the Skeletal Muscle Biobank of the University of Kentucky (n=8), and the National Institutes of Health (n=13).

Standard protocol approvals and patient consent

This study was approved by the Institutional Review Boards of the National Institutes of Health, the Johns Hopkins, the Clinic, and the Vall d'Hebron Hospitals. Written informed consent was obtained from each participant. All methods were performed in accordance with the relevant guidelines and regulations.

RNA sequencing

Bulk RNAseq was performed on frozen muscle biopsy specimens as previously described.[16–19] Briefly, RNA was extracted with TRIzol (Thermo Fisher Scientific). Libraries were either prepared with the NeoPrep system according to the TruSeqM Stranded mRNA Library Prep protocol (Illumina, San Diego, CA), or with the NEBNext Poly(A) mRNA Magnetic Isolation Module and Ultra™ II Directional RNA Library Prep Kit for Illumina (New England BioLabs, ref. #E7490 and #E7760).

Single-nuclei RNA-sequencing

For the nuclei isolation, we used a modification of the sucrose-gradient ultracentrifugation nuclei isolation protocol from Schirmer et al.[20] Ten mg of frozen muscle tissue was sectioned and homogenized in 1mL of lysis buffer (0.32M sucrose, 5mM CaCl2, 3mM MgCl2, 0.1mM EDTA, 10mM Tris-HCl pH 8, 1mM DTT, 0.5% Triton X-100 in DEPC-treated water) using 1.4mm ceramic beads low-binding tubes and the Bertin Technology Precellys 24 lysis homogenizer (6500rpm-3times x 30s). The homogenized tissue was transferred into open-top thick-walled polycarbonate ultracentrifuge tubes (25x89 mm, Beckman Coulter) on ice. 3.7mL of sucrose solution (1.8 M sucrose, 3mM MgCl2, 1mM DTT, 10mM Tris-HCl) were pipetted to the bottom of the tube containing lysis buffer generating two separated phases (sucrose on the bottom and homogenate on the top). The tubes were filled almost completely with lysis buffer and weighted for balance. The samples were ultracentrifuged (Beckman Coulter XE-90, SW32 rotor, swinging bucket) at 24,400rpm (107,163rcf) for 2.5 hours at 4ºC, transferred to ice, and the supernatant removed. Two hundred microliters of DEPC water-based PBS were added to each pellet, incubated on ice for 20 minutes, and then pellets were resuspended. The resulting samples were filtered twice using 30μm Miltenyi pre-separation filters. The nuclei were counted using a manual hemocytometer. Between 2000 and 3000 nuclei per sample were loaded in the 10X Genomic Single-Cell

3' system. We performed the 10X nuclei capture and the library preparation protocol according to the manufacturer's instructions without modification.

Single-cell RNA-sequencing

The cell isolation from human muscle biopsies was performed as follows: ~20-25mg of fresh muscle tissue was placed in a 10cm culture dish with 2mL of dissociation buffer (10 ml of Ham-F 10% Horse serum, 1% Penicillin/streptomycin, and 51.28mg of collagenase II (1000U/ml) (Gibco, ref. 1710-015) per sample. Muscle was minced with scissors into ~1-millimeter cubes, placed in 50 mL tubes with 10mL of dissociation buffer, and incubated at 37ºC in a rocking water bath at 70-75rpm for 50 minutes. After the incubation, washing media (Ham-F, 10% Horse serum, 1% penicillin/streptomycin) was added to the samples to bring the volume to 50mL, and these were centrifuged for 5 minutes at 1,500rpm. Forty-two mL of the resulting supernatant was discarded and the remaining volume was used to triturate the pellet with a 5mL pipet. One mL of Collagenase II (Gibco, ref. 17101-015) and 1ml of 11U Dispase (Gibco, ref. 17105-041) was then added to the mixture and incubated at 37ºC in a rocking water bath at 70-75rpm for 15 minutes. Next, the samples were mixed 10 times with a 10mL syringe and a 20G needle. Washing medium was added to bring the volume to 50mL and then the samples were centrifuged for 5 minutes at 1,500 rpm. The resulting pellets were suspended in 10mL of washing media and then filtered through a 70μm strainer. 50mL of washing media was added to each sample and these were centrifuged at 1,500 rpm for 5 minutes. Next, supernatants were aspirated, leaving 300μL of sample which was used for Fluorescent Activated Cell Sorting (FACS). A target capture of 10,000 cells per sample was loaded in the 10X Genomic Single-Cell 3' system. We performed 10X nuclei capture as well as library preparation protocol according to the manufacturer's recommendation without modification.

Culture of differentiating human skeletal muscle myoblasts and treatment with different types of interferon

Normal human skeletal muscle myoblasts (HSMMs) were cultured according to the protocol recommended by the supplier (Lonza). When 80% confluent, the cultures were induced to differentiate into myotubes by replacing the growth medium with differentiation medium (Dulbecco's modified Eagle's medium supplemented with 2% horse serum and L-glutamine). Two plates of cells were harvested before differentiation and then daily for 6 days.

To examine the effect of different types of interferon on complement expression we treated HSMMs daily with 100 U/L and 1000 U/L of IFNA2a (R&D, ref. 11100-1), IFNB1 (PeproTech, ref. 300-02BC), and IFNG (PeproTech, ref. 300-02), respectively, for 7 days. Treated cells were harvested for RNA extraction and subsequent RNA sequencing.

Statistical and bioinformatic analysis

Complement gene lists were obtained from the HUGO Gene Nomenclature Committee (HGNC). For the bulk RNAseq, reads were demultiplexed using bcl2fastq/2.20.0 and preprocessed using fastp/0.21.0. The abundance of each gene was generated using Salmon/1.5.2 and quality control output was summarized using multiqc/1.11. Counts were normalized using the Trimmed Means of M values (TMM) from edgeR/3.34.1 for graphical analysis. Differential expression was performed using limma/3.48.3. For the single-cell and single-nuclei RNAseq, reads were demultiplexed and aligned using cellranger/6.0.1. Then the samples were aggregated, normalized (SCTransform), and integrated (RunHarmony) using Seurat/4.1.0.

Public databases of human macrophages (GSE1925)[21] and fibroblasts (GSE67737, GSE50954)[22,23] treated with different types of interferon were used to explore the effects

of such treatment in the expression of complement.

For visualization purposes, we used both the R and Python programming languages. Graphical analysis of single cell and single nuclei RNAseq data used the functions contained in Seurat/4.1.0. The Benjamini-Hochberg correction was used to adjust for multiple comparisons, and a corrected value of p (q value) ≤ 0.05 was considered statistically significant.

Data Availability

The datasets generated and/or analysed during the current study are available in the Gene Expression Omnibus repository (GSE220915).

RESULTS

Differential expression of complement proteins in the different types of IM

To define the transcriptomic profiles of patients with different types of IM, we performed bulk RNAseq on muscle biopsies from 132 IM patients, including those with DM (n=44), AS (n=18), IMNM (n=54), and IBM (n=16). The DM group included patients with autoantibodies recognizing Mi2 (n=12), NXP2 (n=14), TIF1γ (n=12), and MDA5 (n=6), whereas IMNM group consisted of patients with anti-HMGCR (n=44) and anti-SRP (n= 10) autoantibodies. Thirty-three muscle biopsies from healthy comparators also underwent bulk RNAseq profling.

Analysis of the bulk transcriptomic data revealed that the main complement components C1-C4 and C7 were overexpressed in muscle biopsies from patients with each type of IM compared with healthy comparator tissue (Figure 1, Supplementary Figure 1, Table 1). In each type of IM, the genes encoding C8a, C8b, and C9 were expressed at low or undetectable levels and were not differentially expressed compared to normal muscle (Figure 1, Supplementary Figure 1).

Different types of IM had distinct complement gene expression profiles. For example, C6 was only overexpressed in muscle biopsies from IBM patients whereas C5 was overexpressed in muscle biopsies from patients with AS, IMNM, and IBM but not DM (Figure 1, Supplementary Figure 1). IMNM muscle biopsies had lower expression of C1-C2, and C4 compared to muscle biopsies from patients with other types of IM. In contrast, DM muscle biopsies were notable for increased expression of C4 and decreased expression of C3 and C5 relative to the other types of IM. Finally, IBM muscle biopsies were characterized by increased expression of C5 and C6 (Figure 1, Supplementary Figure 1, Table 2). Several complement activator and regulator genes were also deferentially expressed in the

different types of IM (Supplementary Tables 1-4).

Correlation of complement expression with myositis activity

Next, we analyzed the bulk transcriptomic data to determine whether complement expression levels were correlated with the degree of myositis disease activity. As shown in Figure 2, the local expression of complement was positively correlated with markers of muscle regeneration (NCAM1, MYOG, PAX7, MYH3, and MYH8) and canonical T-cell and monocyte/macrophage markers (CD3E, CD4, CD8A, CD14, and CD68). In contrast, there was a negative correlation between the expression of complement proteins with markers of mature muscle cells (ACTA1, MYH1, and MYH2) (Figure 2).

Of note, two muscle biopsies were available from the same anti-Mi2-positive DM patient. The first biopsy was obtained when the patient had very active muscle disease and the second biopsy was obtained when the patient had minimal myositis activity, 5 months after starting a JAK/STAT inhibitor (i.e., tofacitinib). Consistent with the cross-sectional data, the expression of complement genes markedly declined as the patient's myositis became less active (Supplementary Figure 2).

Different cell types coordinately express various complement genes

Muscle tissue includes many different cell types. To determine which cell types express complement genes, we performed single-cell RNAseq on fresh muscle tissue derived from 3 patients undergoing a muscle biopsy for suspected IBM and three healthy volunteers. As shown in Supplementary Figure 3, cell clusters representing myofibers, satellite cells, myeloid cells, venular endothelial cells, endothelial cells, fibroblasts, fibroadipogenic progenitors (FAP), CD4+ T cells, and CD8+ T cells could be identified. Genes encoding C1QA, C1QB, and C1QC were expressed at the highest levels in CD14+/CD68+ myeloid cells (i.e., macrophages) whereas genes encoding C1R, C1S, and C3 were primarily ex-

pressed in fibroblasts (Figure 3, Supplementary Figure 4). Unlike in healthy muscle, suspected IBM muscle biopsies showed expression of C1R, and C1S in satellite cells and, to a lesser degree, in FAP cells.

To validate the gene expression data from the single-cell experiments, we performed single-nuclei RNAseq using a subset of 15 frozen muscle biopsy specimens. This included biopsies from 4 patients with DM (2 with anti-Mi2 and 2 with anti-NXP2 autoantibodies), 3 patients with anti-Jo1-positive AS, 6 patients with IMNM (4 with anti-HMGCR and 2 with anti-SRP autoantibodies), and 2 patients with IBM. As shown in Supplementary Figure 5, transcriptomic data from single-nuclei could be used to identify clusters of cells representing mature muscle fibers, satellite cells (i.e., muscle cell precursors), endothelial cells, fibroblasts, T cells, myeloid cells, FAP cells, and adipocytes.

Confirming the results of the single-cell experiment, genes encoding C1QA, C1QB, and C1QC were primarily expressed by myeloid cells (Supplementary Figures 6-7). In contrast, the genes encoding C1R, C1S, and C3 were predominantly expressed by fibroblasts. C1R and C1S were expressed by satellite cells and FAP, albeit at lower levels than in fibroblasts (Supplementary Figures 6-7).

Local complement expression in IM correlates with IFNγ pathway activation

As IFNγ is known to induce the expression of several complement genes in cultured muscle cells, macrophages, and fibroblasts,[24–27] and IFNγ-stimulated genes are expressed at high levels in certain types of IM,[17] we studied the association between IFNγ-stimulated gene and complement gene expression in our bulk transcriptomic data. This analysis revealed a strong correlation between the expression of prominent IFNγ-stimulated genes (e.g., IFI30, and GBP2) and complement genes (Figure 4, Supplementary Figure 8).

Next, we sought to determine whether IFNγ or other interferons could stimulate the ex-

pression of complement genes in the complement-expressing cells identified in the single-nuclei and single-cell RNAseq experiments.

In cultured human skeletal muscle cells, C1R, C1S, C2-C5, and C7, were expressed during the differentiation of myoblasts into myotubes, with the largest increase occurring within the first two days of differentiation (Supplementary Figure 9). Furthermore, treatment of human myoblasts with IFNγ markedly increased the baseline expression of C1R, C1S, and C2-C4. Treatment of cultured muscle cells with IFN$\beta 1$ also induced the expression of some complement genes, albeit to a lesser degree. However, treatment of cultured muscle cells with IFNα had little effect on the expression of complement genes (Figure 5). Taken together, these results support the hypothesis that both muscle differentiation and IFNγ stimulate complement gene expression in human myoblasts and myotubes.

We utilized publicly available datasets to determine whether IFNγ modulates complement gene expression in cultured human macrophages and fibroblasts. Indeed, human macrophages treated with IFNγ showed a marked increase in the expression of C1R, C1S, C2, and especially C1QB (Supplementary Figure 10). Similarly, fibroblasts express C1R and C1S at high levels and treatment with IFNγ results in a small, but consistent, increase in the expression of these two genes (Supplementary Figures 11-12).

DISCUSSION

In this study, we used bulk transcriptomic data from human muscle biopsies to demonstrate that various complement genes are expressed locally within muscle tissue and that local complement expression levels correlate with IM disease activity. The bulk transcriptomic data also revealed that muscle from each type of IM has a distinct "complement expression signature". We then used single-nuclei and single-cell RNAseq techniques to show that macrophages, fibroblasts, and satellite cells are the primary cell types within the muscle that express complement genes. Moreover, we showed that macrophages predominantly express certain complement genes (i.e., C1QA, C1QB, and C1QC) whereas fibroblasts (i.e., C1R, C1S, and C3), satellite cells (i.e., C1R, and C1S), and, to a lower extent, fibroadipocytes (i.e., C1R, and C1S), express a different set of complement genes. Finally, we showed that the expression of complement genes is highly correlated with the expression of IFNγ-stimulated genes in IM muscle and that IFNγ induces complement overexpression in differentiating human skeletal muscle myoblasts, macrophages, and, at least to some degree, in fibroblasts. This suggests that the overexpression of complement genes in patients with myositis is not only due to an increased number of cells expressing complement, but also the result of a more intense expression in each of those cells induced by IFNγ. In this regard, it's worth noting that IMNM has the lowest expression levels of both IFNγ-stimulated genes and complement genes.

The pathophysiologic relevance of local complement production has been studied in other inflammatory diseases. For example, C3 expression by synovial fibroblasts has been linked to inflammation-mediated tissue priming in arthritis[28]. Furthermore, analogous to what we have shown here for IM muscle, different complement genes are differentially expressed by different cells of the lung[29]. Specifically, lung macrophages express C1QA, C1QB, and C1QC, whereas lung mesothelial cells and fibroblasts expressed C1R, C1S, and C3[29]. Taken together, these findings suggest that the coordinated local expression

of complement genes is not specific to muscle tissue in IM but, rather, may be a general mechanism to regulate the complement pathway in inflamed tissues.

This paper has several limitations. First, the expression of certain complement genes fell below the detection threshold of the sequencing techniques that we used. For example, C8a and C8b could not be detected in the bulk RNAseq. Moreover, several genes, including C4a and C4b, were detected by bulk RNAseq but could not be identified by single-nuclei or single-cell RNAseq. Second, we used only RNA-based sequencing methods to study the local expression of complement, because it is selective for locally-synthesized complement genes. While this has numerous advantages, including the application of single-nuclei and single-cell sequencing technologies, we cannot estimate the efficiency of the local translation of complement genes from RNA to protein. Finally, we restricted our analysis to the most common types of IM and the less common types of IM may have different patterns of complement expression.

In summary, this transcriptomic analysis has revealed that macrophages, fibroblasts, and satellite cells express complement genes in a complex and highly coordinated manner within IM muscle biopsies. We also provide evidence that the local expression of complement genes may be regulated by IFNγ, a cytokine already known to be a key player in IM pathogenesis. Future studies will be required for a more complete understanding of the pathophysiologic role of the complement system in myositis muscle and other inflamed tissues.

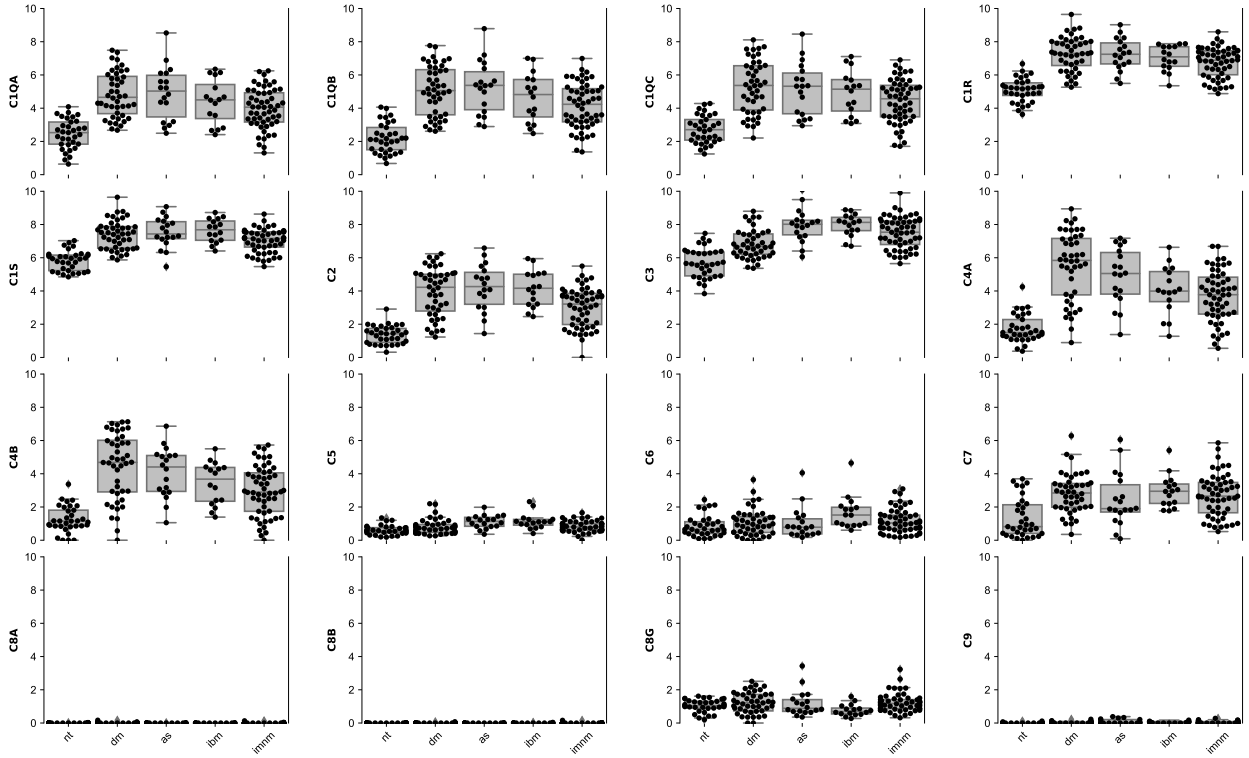
FIGURES AND TABLES

Table 1. Expression of complement genes in the different types of inflammatory myopathy compared to normal muscle.

Gene	DM		AS		IMNM		IBM	
	log2FC	q-value	log2FC	q-value	log2FC	q-value	log2FC	q-value
C1QA	2.7	2e-08	2.7	1e-07	1.8	1e-06	2.2	2e-06
C1QB	3.3	7e-09	3.4	2e-09	2.2	3e-07	2.8	1e-07
C1QC	2.8	7e-08	2.6	6e-08	1.9	3e-07	2.4	1e-07
C1R	2.2	4e-13	2.2	2e-10	1.7	1e-11	2.0	5e-09
C1S	1.6	4e-11	1.7	6e-09	1.2	3e-11	1.9	1e-09
C2	3.3	2e-08	3.5	6e-10	2.3	7e-08	3.5	5e-11
C3	1.3	3e-06	2.3	3e-09	1.8	1e-11	2.3	8e-09
C4A	4.4	7e-08	3.7	5e-09	2.5	4e-06	2.8	5e-07
C4B	4.4	3e-07	3.8	5e-08	2.6	3e-05	3.1	9e-07
C5	0.4	0.2	1.1	1e-04	0.6	0.002	1.2	3e-05
C6	0.3	0.5	0.3	0.5	0.7	0.08	1.7	3e-04
C7	2.7	2e-06	1.9	0.009	2.3	6e-05	2.7	6e-06
C8G	0.4	0.2	0.2	0.6	0.3	0.1	-0.4	0.2

These genes did not pass the cutoff for differential expression: C8A, C8B, C9. *DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis*

Figure 1. Expression of complement genes (log2[TMM+1]) in normal muscle and in different types of inflammatory myopathy. The initial components of the complement cascade, C1-C4, were expressed at the highest levels in each type of myositis. C7 was expressed at an intermediate level, whereas C5, C6, and C8G were expressed at relatively low levels in myositis muscles. Genes encoding C5, C6, C8a, C8b, and C9 were expressed at very low or undetectable levels. Compared with other types of IM, biopsies from immune-mediated necrotizing myopathy patients had lower local levels of complement expression. Graphs were scaled to maximum value of all genes.



nt: normal tissue; dm: dermatomyositis; as: antisynthetase syndrome; imnm: immune-mediated necrotizing myositis; ibm: inclusion body myositis.

Table 2. Expression of complement genes in each group compared to the other types of inflammatory myopathy.

Gene	DM		AS		IMNM		IBM	
	log2FC	q-value	log2FC	q-value	log2FC	q-value	log2FC	q-value
C1QA	0.6	0.1	0.5	0.5	-0.7	0.04	-0.1	0.9
C1QB	0.6	0.1	0.7	0.4	-0.9	0.02	0.0	1
C1QC	0.6	0.1	0.4	0.7	-0.7	0.06	0.0	1
C1R	0.3	0.2	0.3	0.7	-0.5	0.05	0.0	1
C1S	0.1	0.7	0.3	0.6	-0.4	0.03	0.4	0.2
C2	0.5	0.2	0.7	0.4	-1.0	0.005	0.6	0.3
C3	-0.8	6e-04	0.6	0.2	0.1	0.8	0.6	0.1
C4A	1.5	7e-04	0.4	0.8	-1.4	0.005	-0.6	0.6
C4B	1.4	0.002	0.4	0.8	-1.4	0.006	-0.4	0.7
C5	-0.5	0.007	0.6	0.2	0.0	0.9	0.6	0.04
C6	-0.5	0.3	-0.3	0.8	0.1	0.9	1.2	0.03
C7	0.3	0.5	-0.6	0.6	-0.1	0.8	0.3	0.7
C8G	0.2	0.4	0.0	1	0.2	0.6	-0.8	0.06

These genes did not pass the cutoff for differential expression: C8A, C8B, C9. *DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis*

Figure 2. Correlation of complement expression with myositis disease activity. The analysis of bulk transcriptomic data showed that the local expression of complement was positively correlated with markers of muscle regeneration (NCAM1, MYOG, PAX7, MYH3, and MYH8) and canonical T-cell and monocyte/macrophage markers (CD3E, CD4, CD8A, CD14, and CD68). In contrast, there was a negative correlation between the expression of complement proteins with markers of mature muscle cells markers (ACTA1, MYH1, and MYH2).

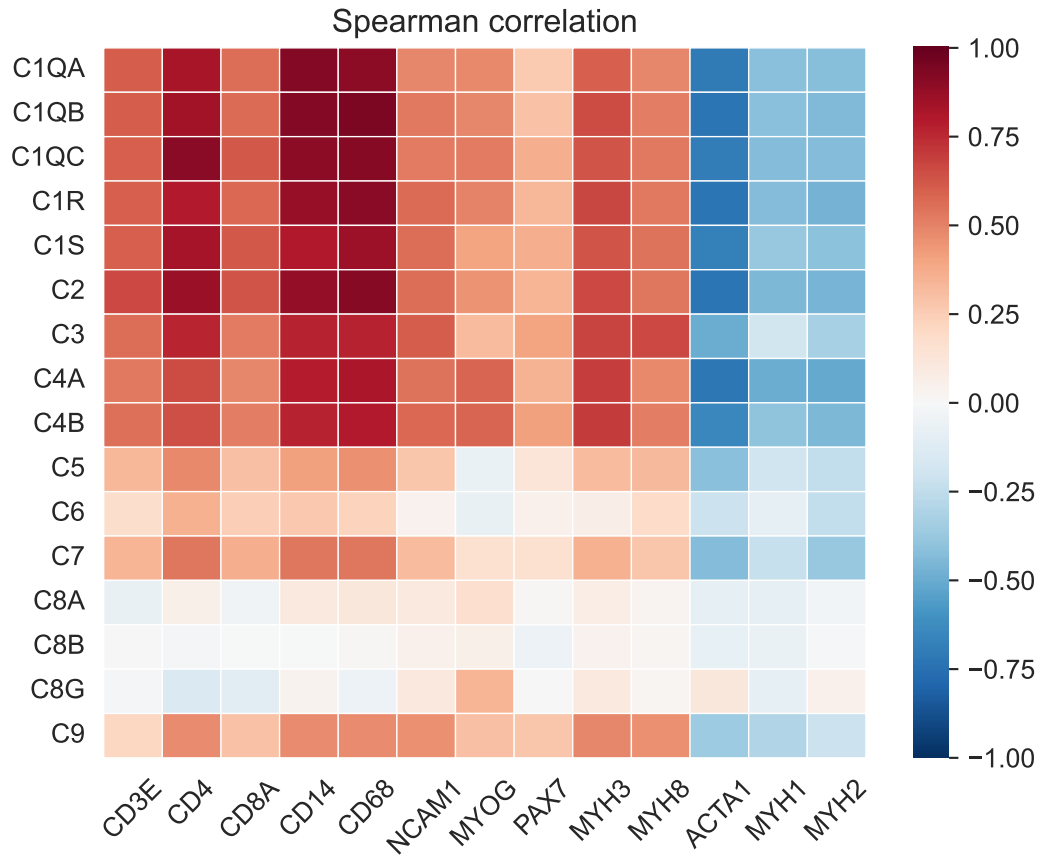
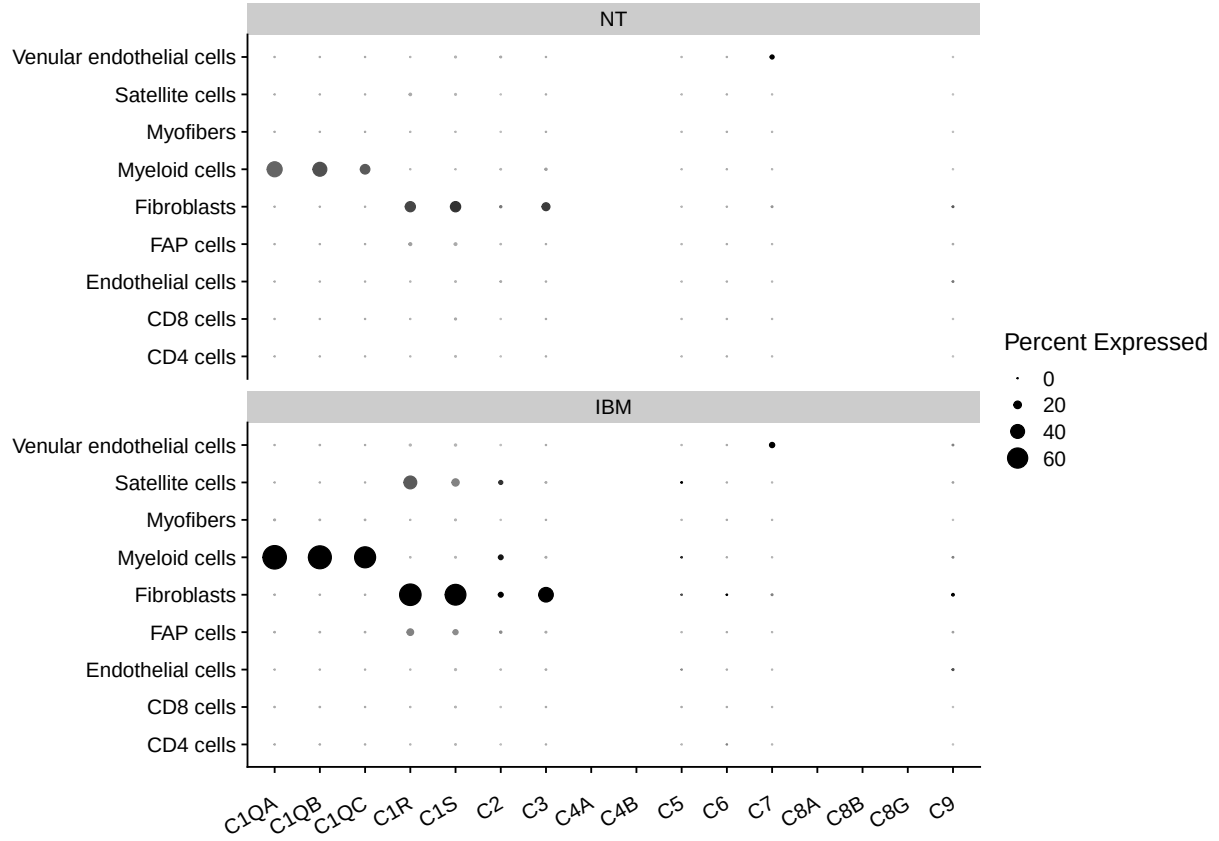


Figure 3. Single-cell RNA sequencing analysis of complement genes from fresh muscle tissue. Biopsies from 3 patients with suspected IBM and 3 healthy volunteers were included. C1QA, C1QB, and C1QC were expressed at the highest levels in CD14+/CD68+ myeloid cells (i.e., macrophages) whereas genes encoding C1R, C1s, and C3 were primarily expressed in fibroblasts. Unlike in healthy muscle, IBM muscle biopsies showed expression of C1R, and C1S in satellite cells and, to a lesser degree, in FAP cells.

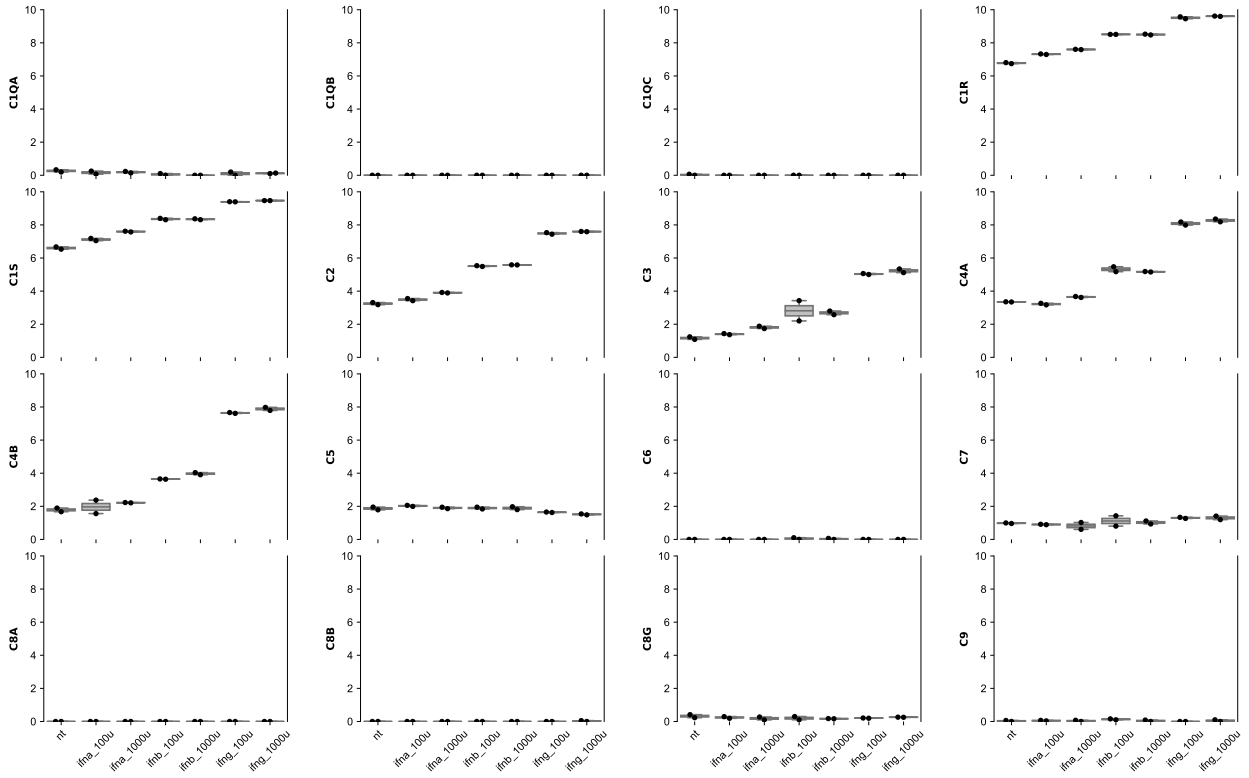


NT: normal tissue; IBM: inclusion body myositis; FAP: fibroadipogenic progenitors.

Figure 4. Correlation of IFI30 with complement genes in normal muscle and different types of inflammatory myopathy. The expression of IFI30, an IFNg-stimulated gene, strongly correlates with the expression of the initial components of the complement cascade.

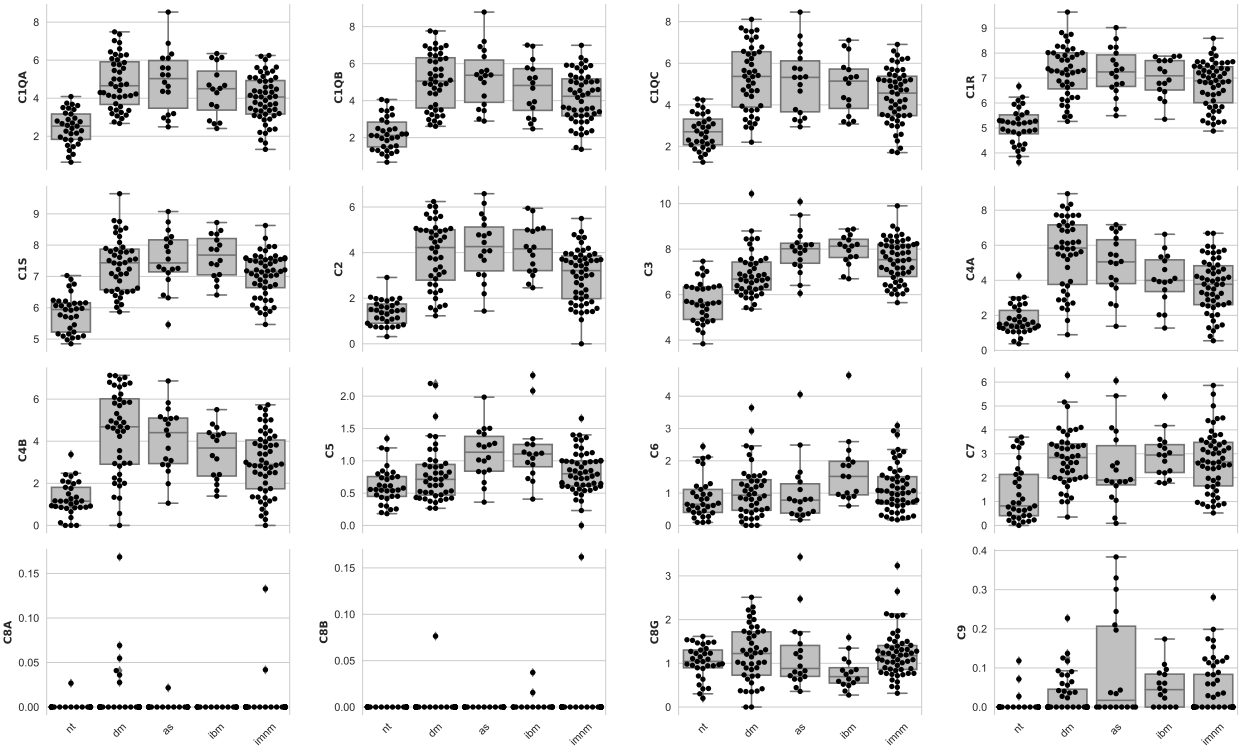
nt: normal tissue; dm: dermatomyositis; as: antisynthetase syndrome; imnm: immune-mediated necrotizing myositis; ibm: inclusion body myositis.

Figure 5. Effect of different types of interferon on complement genes in differentiating human skeletal muscle myoblasts. Increase of the expression (log2[TMM+1]) of C1R, C1S, and C2-C4 with IFNg. Scaled to the maximum value of all genes.



nt: untreated; ifna_100u: treated with 100U of IFNA; ifna_1000u: treated with 1000U of IFNA; ifnb_100u: treated with 100U of IFNB1; ifnb_1000u: treated with 1000U of IFNB1; ifng_100u: treated with 100U of IFNG;. ifng_1000u: treated with 1000U of IFNG.

Supplementary Figure 1. Expression of complement genes (log2[TMM+1]) in normal muscle and in different types of inflammatory myopathy. The initial components of the complement cascade, C1-C4, were expressed at the highest levels in each type of myositis. C7 was expressed at an intermediate level, whereas C5, C6, and C8G were expressed at relatively low levels in myositis muscles. Genes encoding C5, C6, C8a, C8b, and C9 were expressed at very low or undetectable levels. Compared with other types of IM, biopsies from immune-mediated necrotizing myopathy patients had lower local levels of complement expression. Scaled to maximum value of each gene.



nt: normal tissue; dm: dermatomyositis; as: antisynthetase syndrome; imnm: immune-mediated necrotizing myositis; ibm: inclusion body myositis.

Supplementary Table 1. Expression of complement activator genes in the different types of inflammatory myopathy compared to normal muscle.

Gene	DM		AS		IMNM		IBM	
	log2FC	q-value	log2FC	q-value	log2FC	q-value	log2FC	q-value
CFB	3.9	9e-09	3.1	1e-10	2.0	4e-09	2.5	1e-09
CFD	0.5	0.07	0.0	1	0.4	0.05	0.8	0.007
CFP	1.7	7e-06	1.7	4e-05	1.3	6e-05	1.7	2e-05
FCN1	1.8	5e-04	1.9	5e-05	1.6	2e-04	1.9	8e-04
FCN3	1.7	1e-04	0.2	0.7	0.6	0.2	-0.3	0.5
MASP1	-1.6	4e-06	-2.2	3e-05	-1.0	0.001	-0.4	0.4

These genes did not pass the cutoff for differential expression: COLEC10, COLEC11, FCN2, MASP2, MBL2. *DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis*

Supplementary Table 2. Expression of complement activator genes in each group compared to the other types of inflammatory myopathy.

Gene	DM		AS		IMNM		IBM	
	log2FC	q-value	log2FC	q-value	log2FC	q-value	log2FC	q-value
CFB	1.5	2e-06	0.3	0.8	-1.4	7e-05	-0.4	0.6
CFD	0.1	0.9	-0.5	0.4	0.0	1	0.5	0.4
CFP	0.2	0.5	0.3	0.7	-0.4	0.2	0.2	0.7
COLEC11	0.7	0.1	0.2	0.9	-0.5	0.3	-0.5	0.5
FCN1	0.1	0.9	0.3	0.8	-0.2	0.7	0.1	0.9
FCN3	1.3	1e-04	-0.6	0.6	-0.4	0.4	-1.3	0.07
MASP1	-0.6	0.1	-0.9	0.3	0.5	0.2	1.1	0.04

These genes did not pass the cutoff for differential expression: COLEC10, FCN2, MASP2, MBL2. *DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis*

Supplementary Table 3. Expression of complement regulator genes in the different types of inflammatory myopathy compared to normal muscle.

Gene	DM		AS		IMNM		IBM	
	log2FC	q-value	log2FC	q-value	log2FC	q-value	log2FC	q-value
C3AR1	2.1	3e-06	2.5	4e-07	2.0	3e-06	2.3	9e-07
C5AR1	2.2	2e-07	2.3	9e-08	1.9	8e-08	1.6	3e-07
CD46	-0.5	8e-04	-0.4	0.001	-0.4	2e-06	0.1	0.4
CD55	-0.2	0.3	-0.2	0.08	-0.1	0.5	-0.3	0.05
CD59	-0.3	0.002	-0.5	5e-05	-0.2	0.005	-0.2	0.07
CD93	-0.3	0.07	0.1	0.4	0.0	0.7	0.2	0.1
CFH	1.3	2e-06	1.8	2e-07	1.4	6e-09	1.8	4e-08
CFHR1	3.3	3e-04	4.4	8e-06	2.9	0.005	1.9	0.02
CFHR3	1.2	0.03	2.3	5e-05	1.0	0.04	1.9	6e-04
CFI	0.9	0.002	1.1	3e-04	0.8	5e-05	1.1	4e-07
CLU	1.2	8e-06	1.1	0.003	1.0	2e-04	1.4	2e-04
CR1	0.9	0.08	2.8	6e-09	2.0	2e-05	2.4	1e-05
CSMD1	1.0	0.01	1.0	0.06	1.8	3e-07	1.0	0.02
ELANE	0.6	0.07	-0.2	0.6	0.5	0.1	0.5	0.2
ITGAM	1.1	6e-04	1.8	2e-08	1.3	2e-07	1.9	2e-07
ITGAX	1.4	6e-04	1.8	2e-05	1.7	5e-07	2.0	7e-05
ITGB2	1.6	4e-05	2.4	8e-08	1.7	2e-07	2.3	8e-07
SERPING1	1.6	3e-11	1.3	8e-07	1.0	5e-09	1.4	5e-08
VSIG4	2.2	3e-07	2.7	6e-08	2.2	1e-07	2.2	3e-08
VTN	-0.3	0.5	-1.3	0.02	-0.3	0.4	-0.5	0.3

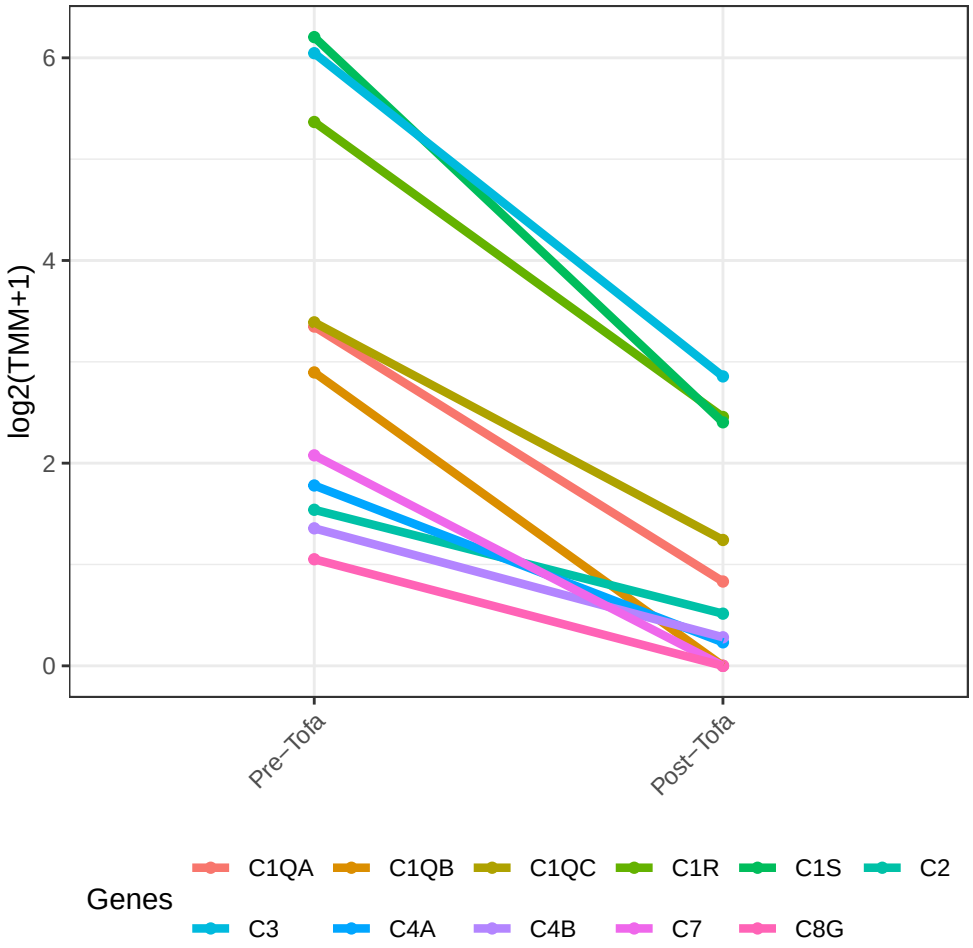
These genes did not pass the cutoff for differential expression: C4BPA, C4BPB, CFHR2, CFHR4, CFHR5, CR2, CSMD2, CSMD3, F2. *DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis*

Supplementary Table 4. Expression of complement regulator genes in each group compared to the other types of inflammatory myopathy.

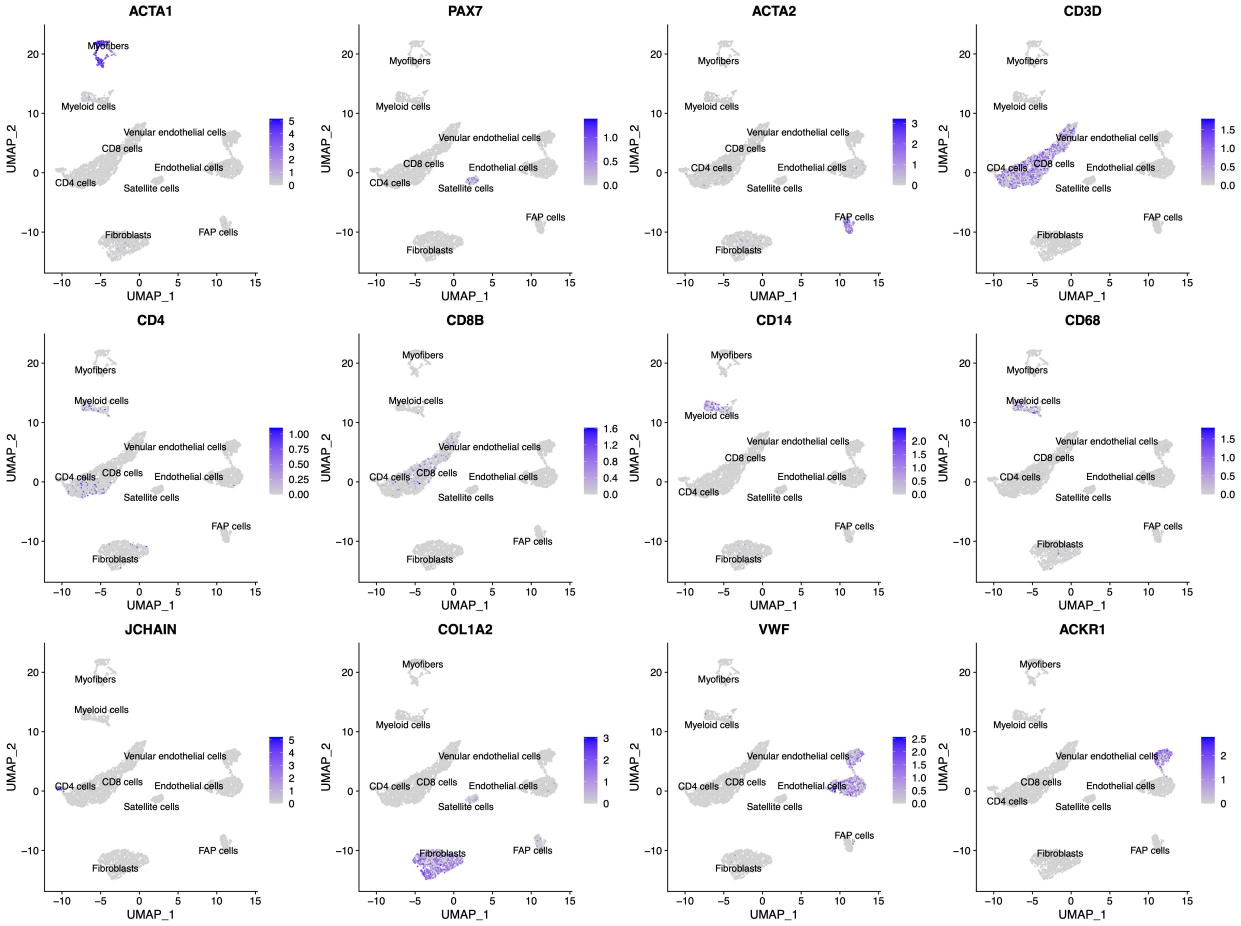
Gene	DM		AS		IMNM		IBM	
	log2FC	q-value	log2FC	q-value	log2FC	q-value	log2FC	q-value
C3AR1	-0.1	0.7	0.5	0.5	-0.2	0.6	0.3	0.7
C4BPB	-1.0	0.02	1.4	0.03	0.1	0.9	0.5	0.5
C5AR1	0.2	0.6	0.4	0.5	-0.2	0.7	-0.5	0.4
CD46	-0.2	0.06	0.0	1	0.0	1	0.5	0.008
CD55	-0.1	0.6	-0.1	0.9	0.2	0.3	-0.1	0.6
CD59	-0.1	0.2	-0.2	0.4	0.2	0.09	0.1	0.6
CD93	-0.5	9e-04	0.3	0.5	0.2	0.3	0.3	0.3
CFH	-0.3	0.2	0.4	0.4	-0.1	0.8	0.4	0.3
CFHR1	0.3	0.7	1.6	0.2	-0.4	0.6	-1.4	0.3
CFHR3	-0.3	0.5	1.2	0.1	-0.6	0.2	0.7	0.3
CFI	-0.1	0.7	0.3	0.7	-0.1	0.7	0.2	0.6
CLU	0.1	0.7	0.0	1	-0.2	0.4	0.3	0.5
CR1	-1.4	2e-04	1.2	0.05	0.4	0.4	0.8	0.2
CSMD1	-0.6	0.07	-0.3	0.7	0.8	0.003	-0.3	0.6
ELANE	0.2	0.6	-0.6	0.5	0.1	0.9	0.0	1
ITGAM	-0.5	0.04	0.6	0.3	0.0	0.9	0.6	0.1
ITGAX	-0.5	0.2	0.3	0.8	0.1	0.8	0.4	0.5
ITGB2	-0.4	0.2	0.7	0.3	-0.2	0.6	0.6	0.2
SERPING1	0.4	0.03	0.0	1	-0.4	0.01	0.1	0.7
VSIG4	-0.2	0.6	0.6	0.4	-0.1	0.9	0.0	0.9
VTN	0.2	0.7	-0.9	0.3	0.3	0.5	-0.1	0.9

These genes did not pass the cutoff for differential expression: C4BPA, CFHR2, CFHR4, CFHR5, CR2, CSMD2, CSMD3, F2. *DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis*

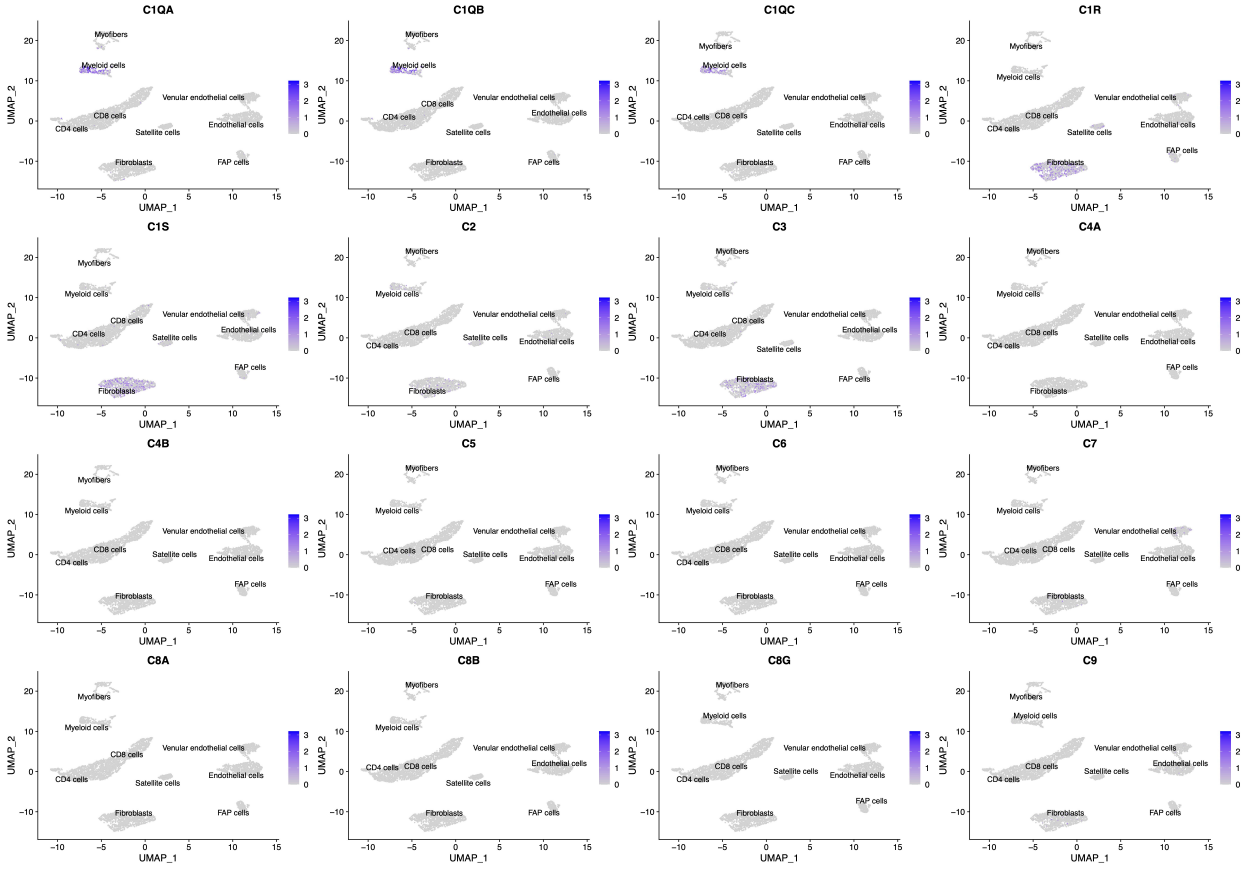
Supplementary Figure 2. Complement gene expression levels before and after treatment with tofacitinib (Tofa) in an anti-Mi2-positive dermatomyositis patient.



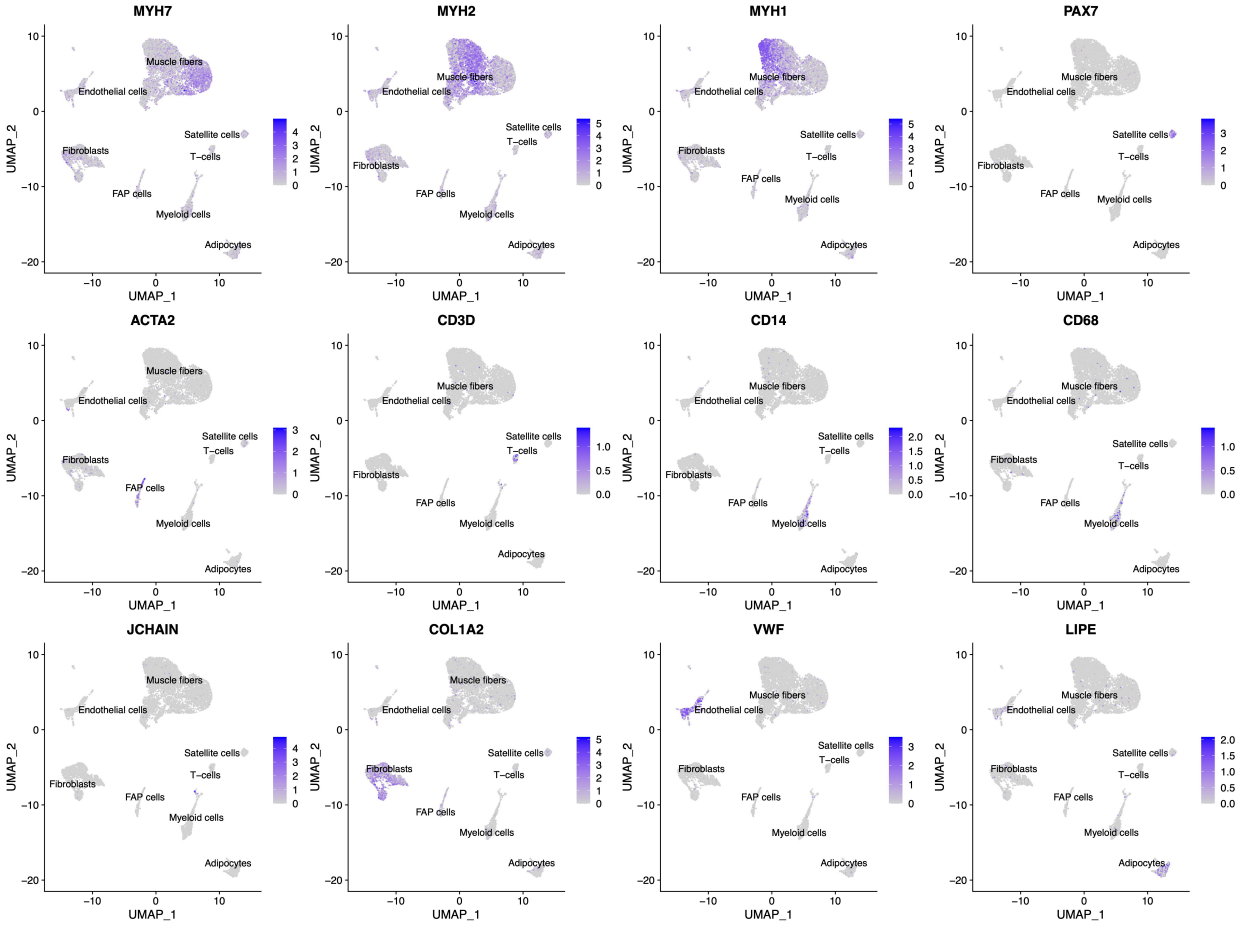
Supplementary Figure 3. Representative genes in each cluster by single-cell RNAseq.



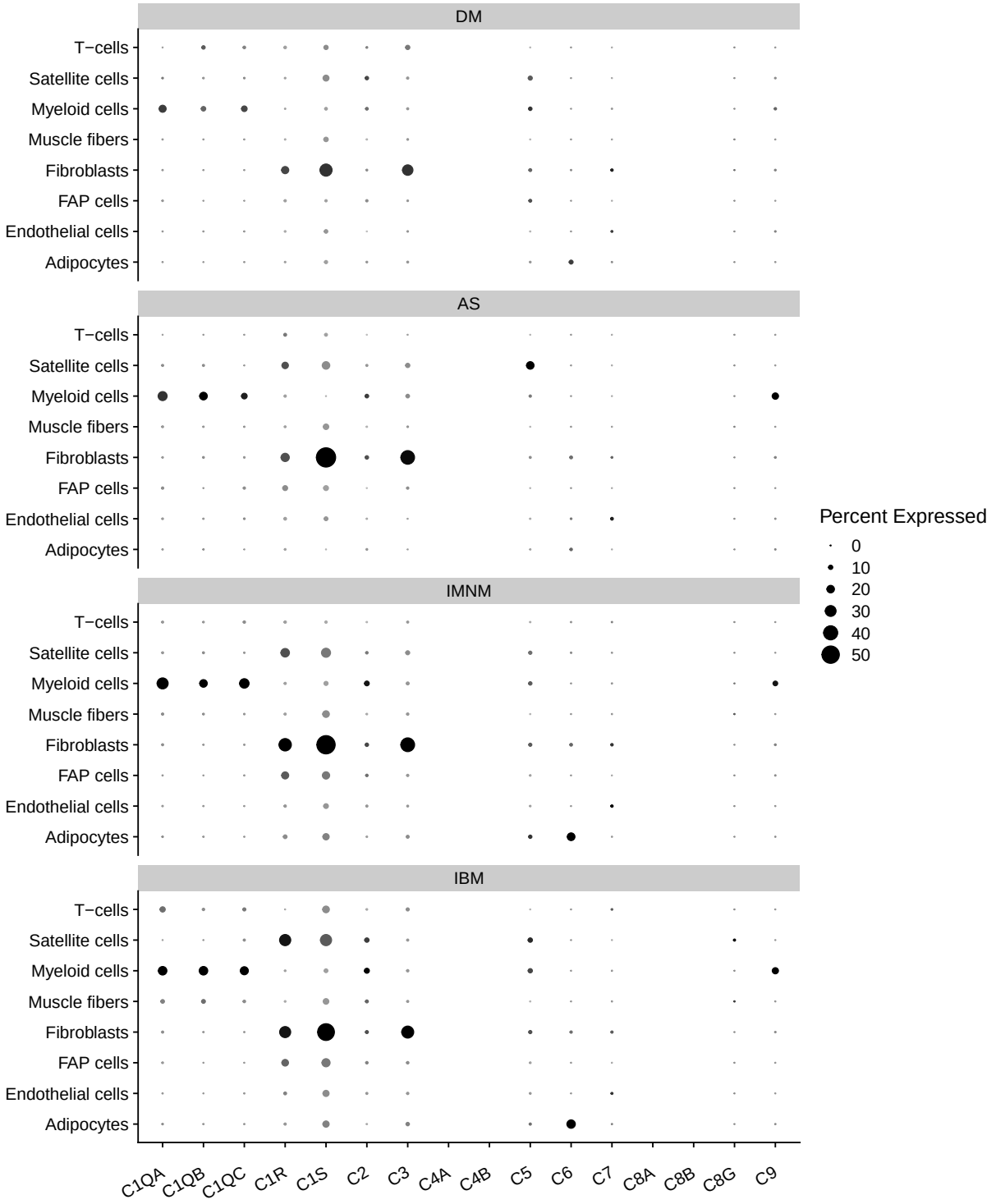
Supplementary Figure 4. Single-cell RNAseq analysis of the expression of complement genes in different muscle cells from fresh muscle tissue. Samples derived from biopsies of 3 patients with a suspected IBM and 3 healthy volunteers were included. Genes encoding C1QA, C1QB, and C1QC were expressed at the highest levels in CD14+/CD68+ myeloid cells (i.e., macrophages) whereas genes encoding C1r, C1s, and C3 were primarily expressed in fibroblasts.



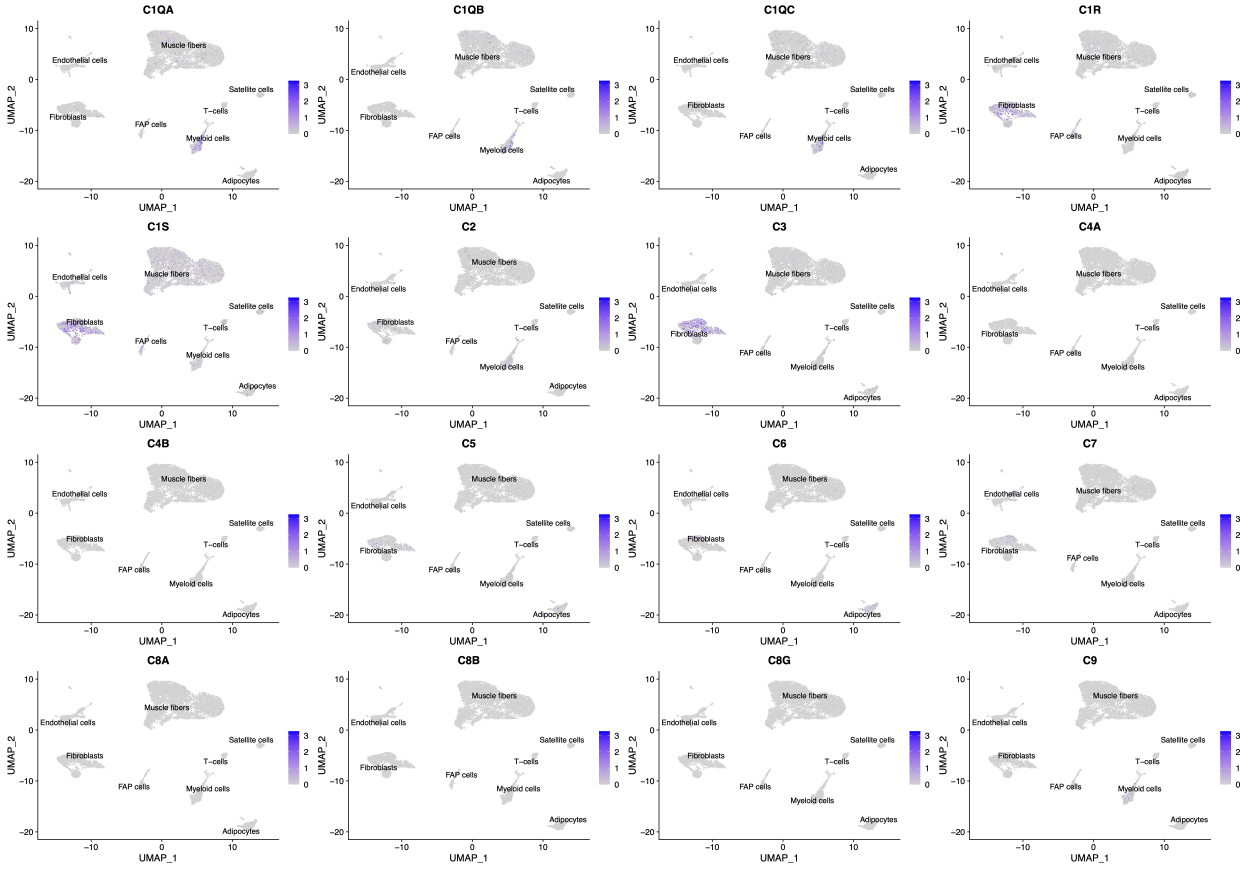
Supplementary Figure 5. Representative genes in each cluster by single-nuclei RNAseq.



Supplementary Figure 6. Expression of complement genes in different muscle cells and different types of inflammatory myopathy by single-nuclei RNAseq.



Supplementary Figure 7. Expression of complement genes in different muscle cells by single-nuclei RNAseq. Single nuclei RNAseq data form 15 frozen muscle biopsies (4 patients with DM, 3 patients with anti-Jo1-positive AS, 6 patients with IMNM, and 2 patients with IBM) was analyzed to determine which cell types express complement genes. Genes encoding C1qA, C1qB, and C1qC were primarily expressed by myeloid cells, whereas genes encoding C1R, C1s, and C3 were primarily expressed in fibroblasts.

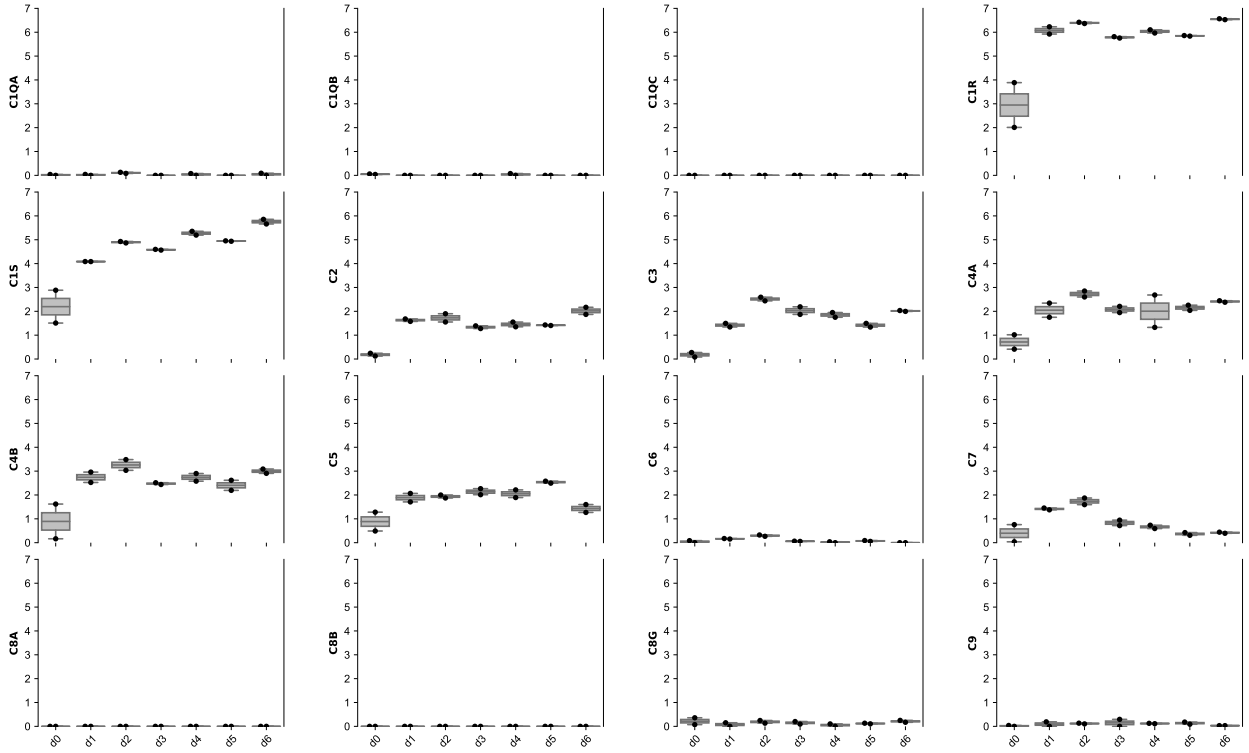


DM: dermatomyositis; AS: antisynthetase syndrome; IMNM: immune-mediated necrotizing myositis; IBM: inclusion body myositis.

Supplementary Figure 8. Correlation of GBP2 with complement genes in normal muscle and in different types of inflammatory myopathy. The expression of GBP2, an IFNg-stimulated gene, strongly correlates with the expression of the initial components of the complement cascade.

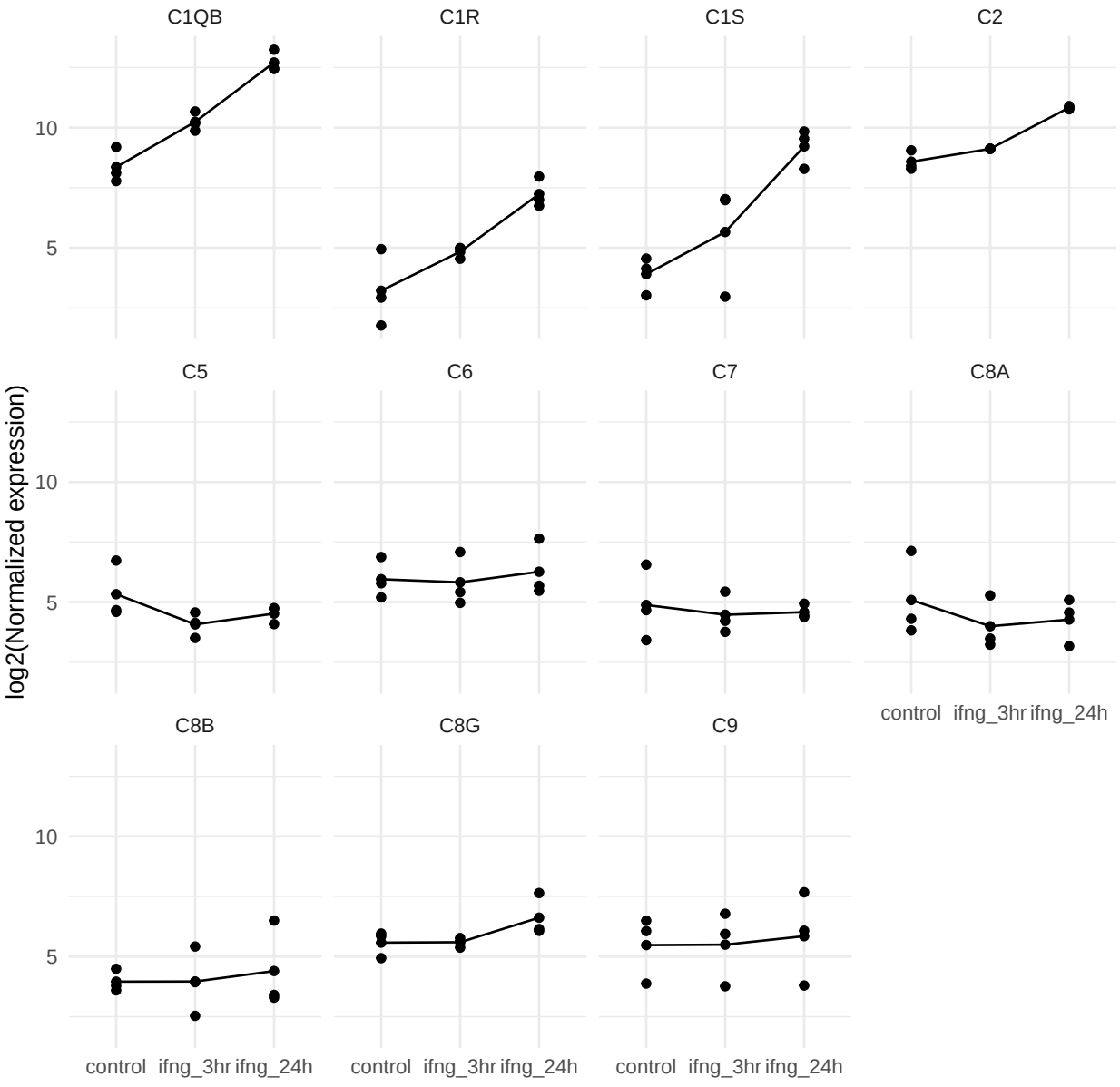
nt: normal tissue; dm: dermatomyositis; as: antisynthetase syndrome; imnm: immune-mediated necrotizing myositis; ibm: inclusion body myositis.

Supplementary Figure 9. Expression of complement genes (log2[TMM+1]) in differentiating human skeletal muscle myoblasts. Overexpression of various complement genes, most predominantly C1R and C1S, early after starting differentiation. Scaled to the maximum value of all genes.



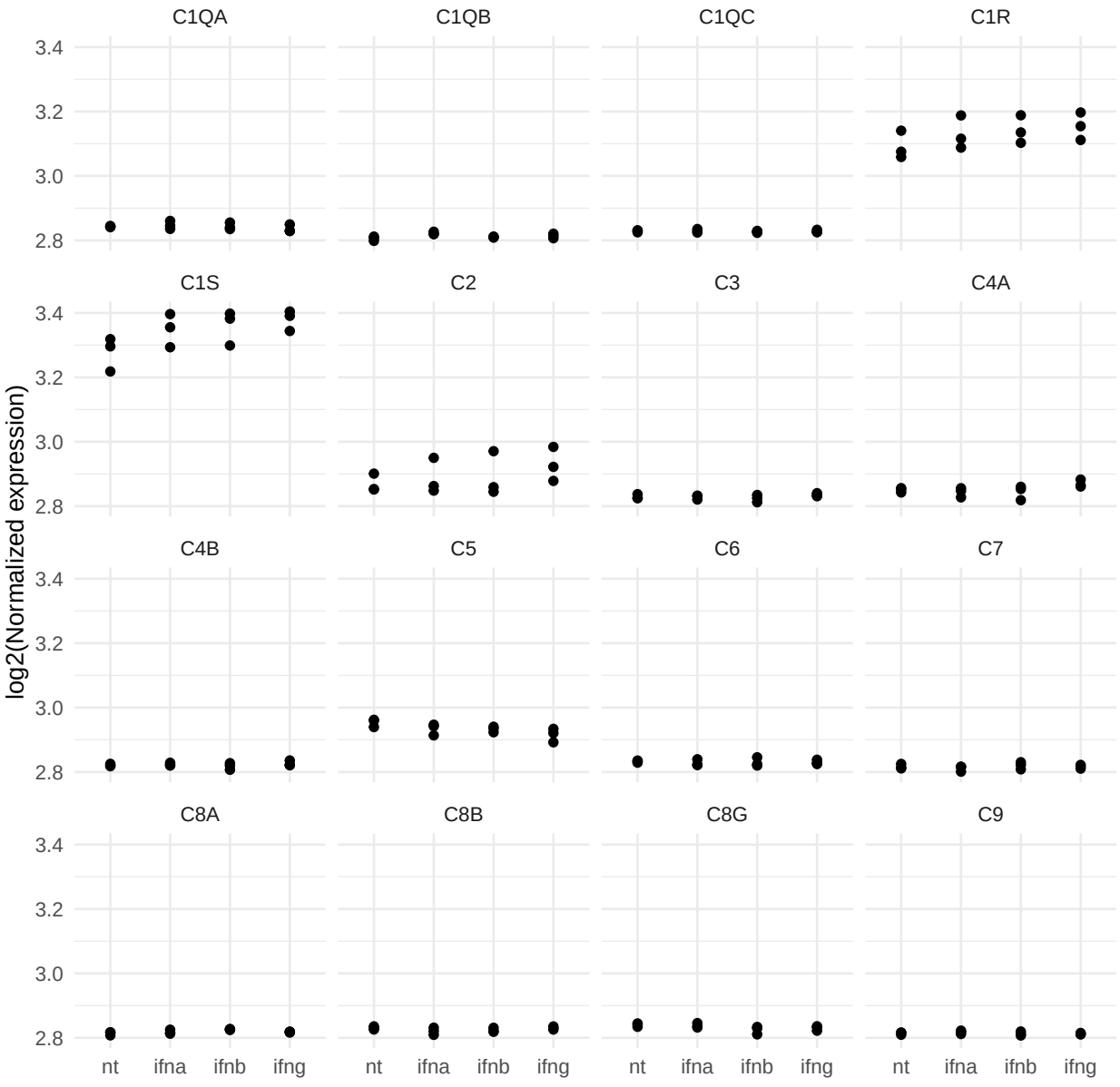
d0: before start of differentiation media; d1-d6: days after starting the differentiation media.

Supplementary Figure 10. Effect of interferon gamma on complement genes in macrophages from GSE1925. Macrophages treated with IFNg overexpressed the initial components of the complement pathway (C1QB, C1R, C1S, C2). C1QB had the highest detectable expression among complement genes.



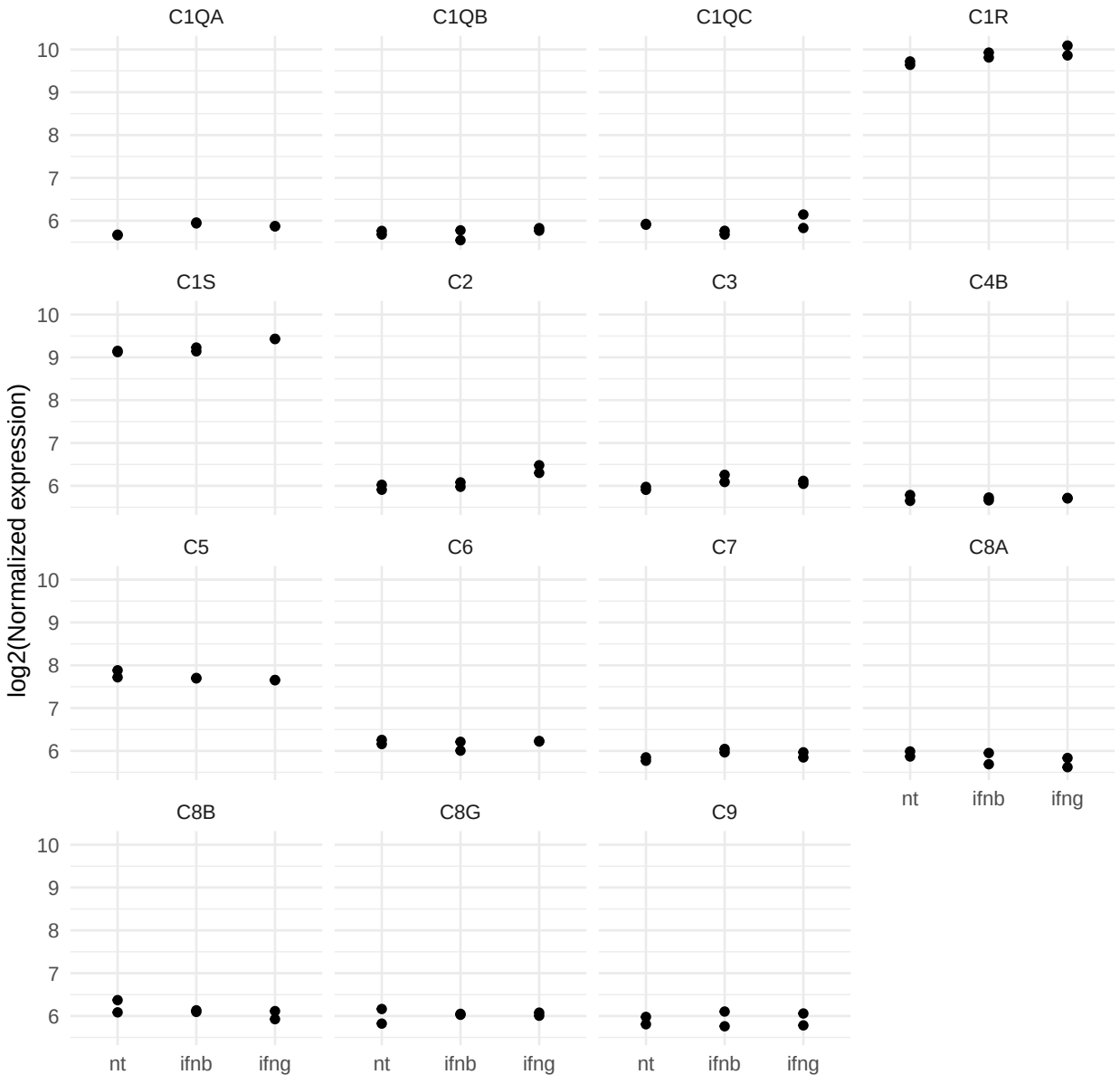
Scaled to the maximum value of all genes. control: untreated macrophages; ifng_3hr: treatment with 100 U/ml of interferon gamma for 3 hours: ifng_24h: treatment with 100 U/ml of interferon gamma for 24 hours. C1QA, C1QC, C3, C4A, C4B were not available in this dataset.

Supplementary Figure 11. Effect of different types of interferon on complement gene expression in fibroblasts (GSE67737). C1R and C1S had the highest expression among complement genes. There was a discreet overexpression of C1R and C1S after treatment with IFNg.



Scaled to the maximum value of all genes. nt: untreated; ifna: treated with 1000IU/mL of IFNA for 10h; ifnb_100u: treated with 1000IU/mL of IFNB for 10h; ifng_100u: treated with 1000IU/mL of IFNG for 10h.

Supplementary Figure 12. Effect of different types of interferon on complement gene expression in fibroblasts (GSE50954). C1R and C1S had the highest expression among complement genes. There was a discreet overexpression of C1R and C1S after treatment with IFNg.



Scaled to the maximum value of all genes. nt: untreated; ifnb: treated with IFNB for 6h; ifng: treated with IFNG for 6h.

REFERENCES

1. Zipfel, P. F. & Skerka, C. Complement regulators and inhibitory proteins. *Nat Rev Immunol* **9**, 729–40 (2009).

2. Morgan, B. P. & Gasque, P. Extrahepatic complement biosynthesis: Where, when and why? *Clinical and experimental immunology* **107**, 1–7 (1997).

3. Cho, H. Complement regulation: Physiology and disease relevance. *Korean J Pediatr* **58**, 239–44 (2015).

4. Selva-O'Callaghan, A. *et al.* Classification and management of adult inflammatory myopathies. *The Lancet. Neurology* **17**, 816–828 (2018).

5. Cong, L. *et al.* [Role of C5b-9 expression in skeletal muscle blood vessels in necrotizing myopathy]. *Nan Fang Yi Ke Da Xue Xue Bao* **32**, 714–7 (2012).

6. Cong, L., Pu, C. Q., Shi, Q., Wang, Q. & Lu, X. H. Complement membrane attack complex is related with immune-mediated necrotizing myopathy. *Int J Clin Exp Pathol* **7**, 4143–9 (2014).

7. Kissel, J. T., Mendell, J. R. & Rammohan, K. W. Microvascular deposition of complement membrane attack complex in dermatomyositis. *N Engl J Med* **314**, 329–34 (1986).

8. Dalakas, M. C. *et al.* A controlled trial of high-dose intravenous immune globulin infusions as treatment for dermatomyositis. *N Engl J Med* **329**, 1993–2000 (1993).

9. Emslie-Smith, A. M. & Engel, A. G. Necrotizing myopathy with pipestem capillaries, microvascular deposition of the complement membrane attack complex (MAC), and minimal cellular infiltration. *Neurology* **41**, 936–9 (1991).

10. Miller, T., Al-Lozi, M. T., Lopate, G. & Pestronk, A. Myopathy with antibodies to the signal recognition particle: Clinical and pathological features. *J Neurol Neurosurg Psychiatry* **73**, 420–8 (2002).

11. De Bleecker, J., Vervaet, V. & Van den Bergh, P. Necrotizing myopathy with microvascular deposition of the complement membrane attack complex. *Clin Neuropathol* **23**, 76–9 (2004).

12. Day, J., Otto, S., Cash, K. & Limaye, V. Clinical and histological features of immune-mediated necrotising myopathy: A multi-centre south australian cohort study. *Neuromuscul Disord* **30**, 186–199 (2020).

13. Paik, J. J. *et al.* Study of tofacitinib in refractory dermatomyositis: An open-label pilot study of ten patients. *Arthritis Rheumatol* **73**, 858–865 (2021).

14. Lloyd, T. E. *et al.* Evaluation and construction of diagnostic criteria for inclusion body myositis. *Neurology* **83**, 426–433 (2014).

15. Casal-Dominguez, M. *et al.* Performance of the 2017 european alliance of associations for rheumatology/american college of rheumatology classification criteria for idiopathic inflammatory myopathies in patients with myositis-specific autoantibodies. *Arthritis & rheumatology (Hoboken, N.J.)* **74**, 508–517 (2022).

16. Amici, D. R. *et al.* Calcium dysregulation, functional calpainopathy, and endoplasmic reticulum stress in sporadic inclusion body myositis. *Acta Neuropathol Commun* **5**, 24 (2017).

17. Pinal-Fernandez, I. *et al.* Identification of distinctive interferon gene signatures in different types of myositis. *Neurology* **93**, e1193–e1204 (2019).

18. Pinal-Fernandez, I. *et al.* Machine learning algorithms reveal unique gene expression profiles in muscle biopsies from patients with different types of myositis. *Annals of the rheumatic diseases* **79**, 1234–1242 (2020).

19. Pinal-Fernandez, I. *et al.* Myositis autoantigen expression correlates with muscle regeneration but not autoantibody specificity. *Arthritis & rheumatology (Hoboken, N.J.)* **71**, 1371–1376 (2019).

20. Schirmer, L. *et al.* Neuronal vulnerability and multilineage diversity in multiple sclerosis. *Nature* **573**, 75–82 (2019).

21. Hu, X., Park-Min, K.-H., Ho, H. H. & Ivashkiv, L. B. IFN-gamma-primed macrophages exhibit increased CCR2-dependent migration and altered IFN-gamma responses mediated by Stat1. *Journal of immunology (Baltimore, Md. : 1950)* **175**, 3637–3647 (2005).

22. Duncan, C. J. A. *et al.* Human IFNAR2 deficiency: Lessons for antiviral immunity. *Science translational medicine* **7**, 307ra154 (2015).

23. Cheon, H. *et al.* IFNb-dependent increases in STAT1, STAT2, and IRF9 mediate resistance to viruses and DNA damage. *The EMBO journal* **32**, 2751–2763 (2013).

24. Legoedec, J., Gasque, P., Jeanne, J. F. & Fontaine, M. Expression of the complement alternative pathway by human myoblasts in vitro: Biosynthesis of C3, factor b, factor h and factor i. *Eur J Immunol* **25**, 3460–6 (1995).

25. Legoedec, J., Gasque, P., Jeanne, J. F., Scotte, M. & Fontaine, M. Complement classical pathway expression by human skeletal myoblasts in vitro. *Mol Immunol* **34**, 735–41 (1997).

26. Gasque, P., Morgan, B. P., Legoedec, J., Chan, P. & Fontaine, M. Human skeletal myoblasts spontaneously activate allogeneic complement but are resistant to killing. *J Immunol* **156**, 3402–11 (1996).

27. Lappin, D. F., Guc, D., Hill, A., McShane, T. & Whaley, K. Effect of interferon-gamma on complement gene expression in different cell types. *The Biochemical journal* **281 (Pt 2)**, 437–442 (1992).

28. Friščić, J. *et al.* The complement system drives local inflammatory tissue priming by metabolic reprogramming of synovial fibroblasts. *Immunity* **54**, 1002–1021.e10 (2021).

29. Chaudhary, N., Jayaraman, A., Reinhardt, C., Campbell, J. D. & Bosmann, M. A single-cell lung atlas of complement genes identifies the mesothelium and epithelium as prominent sources of extrahepatic complement proteins. *Mucosal immunology* **15**, 927–939 (2022).

2.3. Derepresión transcripcional en dermatomiositis anti-Mi2

- La dermatomiositis anti-Mi2 se caracteriza por una afectación muscular más grave y una necrosis de miofibras más prominente que otros tipos de dermatomiositis.
- Los autoanticuerpos anti-Mi2 se dirigen a las subunidades funcionales del complejo NuRD, un represor transcripcional.
- Las biopsias musculares de pacientes con dermatomiositis anti-Mi2 positivo sobreexpresan de un conjunto de más de 100 genes, algunos de los cuales no suelen expresarse en el músculo esquelético.
- Los niveles de expresión de estos genes se correlacionan con los títulos de autoanticuerpos anti-Mi2 y con marcadores de actividad de la enfermedad.
- Los niveles de expresión de los miembros de este conjunto de genes están mutuamente correlacionados.
- Este estudio confirma que los autoanticuerpos anti-Mi2 definen un subtipo distinto de dermatomiositis.
- Nuestros datos sugieren la posibilidad de que los autoanticuerpos anti-Mi2 puedan ejercer un efecto patogénico al inhibir el complejo NuRD y, posteriormente, desreprimir los genes definidos en este estudio.

Transcriptional derepression in the muscle of patients with dermatomyositis and anti-Mi2 autoantibodies

Iago Pinal-Fernandez[a,b*], Jose C. Milisenda[a,c,d*], Katherine Pak[a*], Maria Casal-Dominguez[a,b], Sandra Munoz-Braceras[a], Jiram Torres-Ruiz[a], Stefania Del Orso[a], Faiza Naz[a], Gustavo Gutierrez-Cruz[a], Yaiza Duque-Jaimez[c], Ana Matas-Garcia[c,d,e], Joan Padrosa[e], Francesc J. Garcia-Garcia[c,d,e], Mariona Guitart-Mampel[c,d,e], Gloria Garrabou[c,d,e], Ernesto Trallero-Araguas[f,g], Brian Walitt[h], Julie J Paik[i], Jemima Albayda[i], Lisa Christopher-Stine[b,i], Thomas E. Lloyd[b], Josep Maria Grau[c,d,e], Albert Selva-O'Callaghan[f,g], and Andrew L. Mammen[a,b,i]

[a]Muscle Disease Unit, National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, Bethesda, MD, USA

[b]Department of Neurology, Johns Hopkins University School of Medicine, Baltimore, MD, USA

[c]Muscle Research Unit, Internal Medicine Service, Hospital Clinic, Barcelona, Spain

[d]Barcelona University, Barcelona, Spain

[e]CIBERER, Barcelona, Spain

[f]Systemic Autoimmune Disease Unit, Vall d'Hebron Institute of Research, Barcelona, Spain

[g]Autonomous University of Barcelona, Barcelona, Spain

[h]National Institute of Nursing Research, National Institutes of Health, Bethesda, MD, USA

[i]Department of Medicine, Johns Hopkins University School of Medicine, Baltimore, MD, USA

* These authors contributed equally to this project.

Address correspondence to: Andrew L. Mammen, M.D., Ph.D., or Iago Pinal-Fernandez, M.D., Ph.D., Ph.D., Muscle Disease Unit, National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, 50 South Drive, Room 1141, Building 50, MSC 8024, Bethesda, MD 20892. E-mail: andrew.mammen@nih.gov or iago.pinalfernandez@nih.gov. Phone: 301-451-1199. Fax: 301-594-0305.

Competing interests: None.

Contributorship: All authors contributed to the development of the manuscript, including interpretation of results, substantive review of drafts, and approval of the final draft for submission.

Acknowledgments: None.

Funding: This study was funded, in part, by the Intramural Research Program of the National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health. This work was also supported by the Peter Buck and the Huayi and Siuling Zhang Discovery Fund.

Ethical approval information: All biopsies were from subjects enrolled in institutional review board (IRB)-approved longitudinal cohorts in the National Institutes of Health, the Johns Hopkins, the Clinic Hospital, or the Vall d'Hebron Hospital.

Data sharing statement: Any anonymized data not published within the article will be shared by request from any qualified investigator.

Patient and public involvement: Patients and/or the public were not involved in the design, conduct, reporting, or dissemination plans of this research.

Keywords: Myositis, RNA-sequencing, NURD complex, anti-Mi2, dermatomyositis

KEY MESSAGES

What is already known about this subject?

- Anti-Mi2 dermatomyositis is characterized by more severe muscle involvement and more prominent myofiber necrosis than other types of dermatomyositis.
- Anti-Mi2 autoantibodies target functional subunits of the NuRD complex, a transcriptional repressor.

What does this study add?

- Muscle biopsies from patients with anti-Mi2-positive dermatomyositis are characterized by the overexpression of a set of more than 100 genes, some of them not usually expressed in the muscle.
- The expression levels of these genes are correlated with the titers of anti-Mi2 autoantibodies and with markers of disease activity.
- The expression levels of the members of this gene set are mutually correlated.

How might this impact on clinical practice?

- This study confirms that anti-Mi2 autoantibodies define a distinct subtype of dermatomyositis.
- Our data suggest the possibility that anti-Mi2 autoantibodies could exert a pathogenic effect by inhibiting the NuRD complex and subsequently derepressing the genes defined in this study.

ABSTRACT

Objectives: Myositis is a heterogeneous family of diseases that includes dermatomyositis (DM), immune-mediated necrotizing myopathy (IMNM), antisynthetase syndrome (AS), and inclusion body myositis (IBM). Myositis-specific autoantibodies define different subtypes of myositis. For example, anti-Mi2 autoantibodies have more severe muscle involvement and prominent necrosis than other DM patients. This study aimed to define the transcriptional profile of muscle biopsies from anti-Mi2-positive DM patients.

Methods: RNA sequencing was performed on muscle biopsies (n=171) from patients with anti-Mi2-positive DM (n=18), DM without anti-Mi2 autoantibodies (n=32), AS (n=18), IMNM (n=54), and IBM (n=16) as well as 33 normal muscle biopsies.

Results: A set of 135 genes involved in a variety of cellular pathways was specifically overexpressed in anti-Mi2-positive DM muscle. This set included genes that are not expressed at detectable levels in skeletal muscle (e.g. SCRT1, KCNJ4). The expression levels of these genes were correlated with the titers of anti-Mi2 autoantibodies and markers of disease activity. Moreover, the expression levels of the members of this gene set were mutually correlated.

Conclusions: Muscle biopsies from patients with anti-Mi2-positive DM are specifically characterized by the coordinated overexpression of a set of more than 100 genes, many not usually expressed in skeletal muscle. Anti-Mi2 titers and markers of disease activity correlated with the expression of this gene set. Importantly, anti-Mi2 autoantibodies target functional subunits of the NuRD complex, a transcriptional repressor. This

suggests the possibility that anti-Mi2 autoantibodies could exert a pathogenic effect by inhibiting the NuRD complex and subsequently derepressing the genes defined in this study.

INTRODUCTION

Myositis is a family of autoimmune systemic disorders affecting not only the muscle, but also other organs and systems, such as the skin, the lungs, and/or the joints. Most myositis patients have a unique myositis-specific autoantibody (MSA). The MSAs so clearly define unique subsets of myositis patients that some authors have hypothesized that these autoantibodies may be causally linked to the pathogenesis of the disease.[1-4] However, the pathogenic role of autoantibodies in myositis patients remains unproven.

Among dermatomyositis (DM) patients, the four most prevalent MSAs are anti-Mi2, anti-NPX2, anti-TIF1g, and anti-MDA5 autoantibodies. Of note, those with anti-Mi2 autoantibodies have the most severe muscle disease, with weaker muscles, higher serum muscle enzyme levels, and more prominent myofiber necrosis than other DM patients. Furthermore, in anti-Mi2-positive patients, autoantibody levels correlate with DM disease activity.[5-8]

Anti-Mi2 autoantibodies recognize chromodomain helicase DNA-binding (CHD) proteins, a family of ATP-dependent chromatin remodelers which are functionally critical subunits of the nucleosome remodeling and deacetylase (NuRD) complex, a well-described transcriptional repressor.[9] Although CHD3 and CHD4 are the most common target antigens tested in anti-Mi2 serologic assays, CHD3, CHD4, and CHD5 share significant sequence homology and are likely recognized by autoantibodies in anti-Mi2-positive DM patients.

In the past, our group identified several specific transcriptional features in muscle biopsies from patients with different types of myositis.[10] For example, we showed that MADCAM1 is uniquely overexpressed in muscle biopsies from anti-Mi2-positive DM patients. In the current study, we have focused our analysis on DM patients with anti-Mi2 autoantibodies, allowing us to show that muscle biopsies from these patients have a unique transcriptional profile, with overexpression of a set of more than one hundred genes. We also show that the expression level of each gene in the set is associated with the level of circulating anti-Mi2 autoantibodies and with markers of disease activity. Furthermore, the expression level of the members of this gene set is mutually correlated. Based on these observations, we hypothesize that disruption of NuRD complex function by intracellular anti-Mi2 autoantibodies could derepress gene expression and explain the observed transcriptional pattern.

METHODS

Patients

In this study, we included all muscle biopsies from patients enrolled in institutional review board-approved (IRB) longitudinal cohorts from the National Institutes of Health in Bethesda, MD; the Johns Hopkins Myositis Center in Baltimore, MD; the Vall d'Hebron Hospital, and the Clinic Hospital in Barcelona if they fulfilled Lloyd's criteria for inclusion body myositis,[11] or they fulfilled the Casal and Pinal criteria for other types of myositis,[1] and were positive for one of the following myositis specific autoantibodies (MSA): anti-Mi2, anti-Jo1, anti-NXP2, anti-TIF1g, anti-MDA5, anti-SRP or anti-HMGCR. Autoantibody testing was performed using one or more of the following techniques: ELISA, immunoprecipitation of proteins generated by *in vitro* transcription and translation (IVTT-IP), line blotting (EUROLINE myositis profile), or immunoprecipitation from ^{35}S-methionine-labeled HeLa cell lysates. Patients were classified as antisynthetase syndrome if they were positive for anti-Jo1 autoantibodies, as DM if they tested positive for anti-Mi2, anti-NXP2, anti-MDA5, or anti-TIF1g, and as immune-mediated necrotizing myositis if they had autoantibodies against SRP or HMGCR. We obtained muscle biopsies from several types of myopathies from the Johns Hopkins Hospital and normal muscle biopsies to use as healthy comparators from the Johns Hopkins Neuromuscular Pathology Laboratory (n=12), the Skeletal Muscle Biobank of the University of Kentucky (n=8), and the National Institutes of Health (n=13).

Standard protocol approvals and patient consent

This study was approved by the Institutional Review Boards of the National Institutes of Health, the Johns Hopkins Myositis Center, the Clinic, and the Vall d'Hebron Hospitals. Written informed consent was obtained from each participant. All methods were performed in accordance with the relevant guidelines and regulations.

Anti-Mi2 autoantibody titers

Quantitative anti-Mi2 autoantibody ELISA was performed as previously described.[5, 6] In short, 96-well ELISA plates were coated overnight at 4°C with 100ng of Mi2b protein (Abcam, ab124864) diluted in PBS. Replicate wells were incubated with phosphate-buffered saline (PBS) alone. After washing the plates, human serum samples, diluted 1:400 in PBS with 0.05% Tween (PBS-T), were added to the wells (1 hour, 37°C). Then, HRP-labeled goat anti-human antibody (Jackson ImmunoResearch 109-036-088; 1:10,000) was added to each well (30 minutes, 37° C). Color development was performed using SureBlue™ peroxidase reagent (KPL) and absorbance at 450 nm was measured. For each sample, the background absorbance from the PBS-coated wells was subtracted from that of the corresponding Mi2-coated wells. Test sample absorbances were normalized from an arbitrary positive anti-Mi2 patient, which was used as a reference serum included in every ELISA. The cutoff for a negative anti-Mi2 autoantibody titer was set at 0.17 arbitrary units (mean absorbance plus 3 standard deviations of a healthy control cohort), as previously reported.[5, 6]

RNA sequencing

Bulk RNAseq was performed on frozen muscle biopsy specimens as previously described.[10, 12-15] Briefly, RNA was extracted with TRIzol (Thermo Fisher Scientific). Libraries were either prepared with the NeoPrep system according to the TruSeqM Stranded mRNA Library Prep protocol (Illumina, San Diego, CA) or with the NEBNext Poly(A) mRNA Magnetic Isolation Module and Ultra™ II Directional RNA Library Prep Kit for Illumina (New England BioLabs, ref. #E7490 and #E7760).

Statistical and bioinformatic analysis

Reads were demultiplexed using bcl2fastq/2.20.0 and preprocessed using fastp/0.21.0. The abundance of each gene was determined using Salmon/1.5.2 and quality control was summarized using multiqc/1.11. Counts were normalized using the Trimmed Means of M values (TMM) from edgeR/3.34.1 for graphical analysis. Differential expression was performed using limma/3.48.3. Enrichment analysis was performed using clusterProfiler/4.6.0 and the Reactome dataset.

For visualization purposes, we used both the R and Python programming languages. The Benjamini-Hochberg correction was used to adjust for multiple comparisons, and a corrected value of p (q value) ≤ 0.05 was considered statistically significant.

RESULTS

A set of genes is specifically overexpressed in patients with anti-Mi2-positive DM

To define the transcriptomic profile of anti-Mi2 myositis muscles, we performed bulk RNA sequencing on RNA obtained from 171 muscle biopsies. This included biopsies from 18 anti-Mi2-positive DM patients, 32 DM patients with other MSAs (14 with anti-NXP2, 12 with anti-TIF1γ, and 6 with anti-MDA5 autoantibodies), 18 AS patients, 54 IMNM patients (44 with anti-HMGCR and 10 with anti-SRP autoantibodies), 16 IBM patients, as well as 33 histologically normal muscle biopsies.

We observed that the most significantly expressed genes in patients with anti-Mi2 dermatomyositis were a) not expressed at detectable levels in non-anti-Mi2 muscle biopsies or in normal skeletal muscle, b) were all overexpressed, and c) were unrelated to the interferon pathway (Figure 1). To formally define this set of genes we calculated the intersection of the differentially overexpressed genes (q-value < 0.05) between the anti-Mi2 DM group and each of the other study groups (Figures 2 and 3, Supplementary Table 1). Of note, no differentially expressed genes were simultaneously under-expressed in anti-Mi2-positive DM muscle compared to each of the other study groups (Supplementary Figure 1).

The set of upregulated genes in anti-Mi2-positive DM muscle biopsies comprised genes involved in a variety of cellular pathways. Accordingly, enrichment analysis using this set of genes showed only a very limited number of pathways, with borderline levels of signification (Supplementary Figure 2).

To explore whether the set of genes was overexpressed in a coordinated manner, we performed a correlation analysis. Indeed, there was a marked positive correlation between the expression level of nearly all the genes in the set. This implies that the transcriptional derepression of these genes is tightly coordinated and raises the possibility of a single causal mechanism (Figure 4).

Overexpression of anti-Mi2-specific genes in muscle correlates with autoantibody titers and with transcriptomic markers of disease activity

We next investigated the relationship between anti-Mi2 autoantibody titers and the expression of the set of genes specifically upregulated in muscle biopsies from patients with these autoantibodies. Among the 18 anti-Mi2-positive DM patients whose muscle biopsies were included in this study, 15 had serum samples available for this study and were tested using a previously established quantitative anti-Mi2 autoantibody ELISA [5, 6]. Remarkably, anti-Mi2 autoantibody titers were robustly correlated with the expression levels of the set of genes specifically upregulated in muscle biopsies from patients with anti-Mi2 autoantibodies (Figure 5).

We next analyzed the relationship between the expression of the anti-Mi2-specific genes and previously established transcriptomic markers of disease activity. Indeed, there was a strong positive correlation between the expression of the set anti-Mi2-specific genes with type 1 interferon-inducible genes (ISG15, MX1), type 2 interferon-inducible genes (GBP2, IFI30), T-cell markers (CD3E, CD4, CD8), macrophage markers (CD14, CD68), and markers of muscle differentiation (NCAM1, MYOG, PAX7,

MYH3, MYH8). Conversely, there was a negative correlation between the expression of the set of anti-Mi2-specific genes with the expression of genes encoding structural proteins found in mature muscle (ACTA1, MYH1, MYH2). Taken together, this data suggests that the derepression of anti-Mi2-specific genes is intimately linked with the intensity of muscle disease in patients with anti-Mi2-positive DM.

DISCUSSION

In this study, we describe a highly coordinated program of transcriptional derepression that occurs specifically in the muscles of DM patients with anti-Mi2 autoantibodies. We identified a set of more than 100 genes whose expression levels were linked to transcriptomic markers of disease severity as well as to the titer of anti-Mi2 autoantibodies. Taken together with previous reports demonstrating that anti-Mi2-positive DM patients have specific clinical features and biopsy characteristics [1, 5, 7, 8], this transcriptional effect suggests that anti-Mi2 autoantibodies are not just a useful biomarker, but are intimately related to disease pathogenesis and define a distinct subtype of DM.

Based on the coordinated derepression of this heterogeneous set of genes, we hypothesize that anti-Mi2 autoantibodies may penetrate the sarcolemma of previously damaged muscle fibers, interact with nascent CHD proteins in the cytoplasm, and interfere with the supply of CHD proteins to form the NURD complex. As the NuRD complex serves to inhibit gene expression, disrupting its function could induce the transcriptional derepression that we observe and may be toxic to muscle fibers. This hypothesis would help explain the presence of necrotic muscle fibers surrounded by healthy muscle fibers, a histologic finding frequently observed in the muscle biopsies of anti-Mi2-positive DM patients.[5, 7, 8]

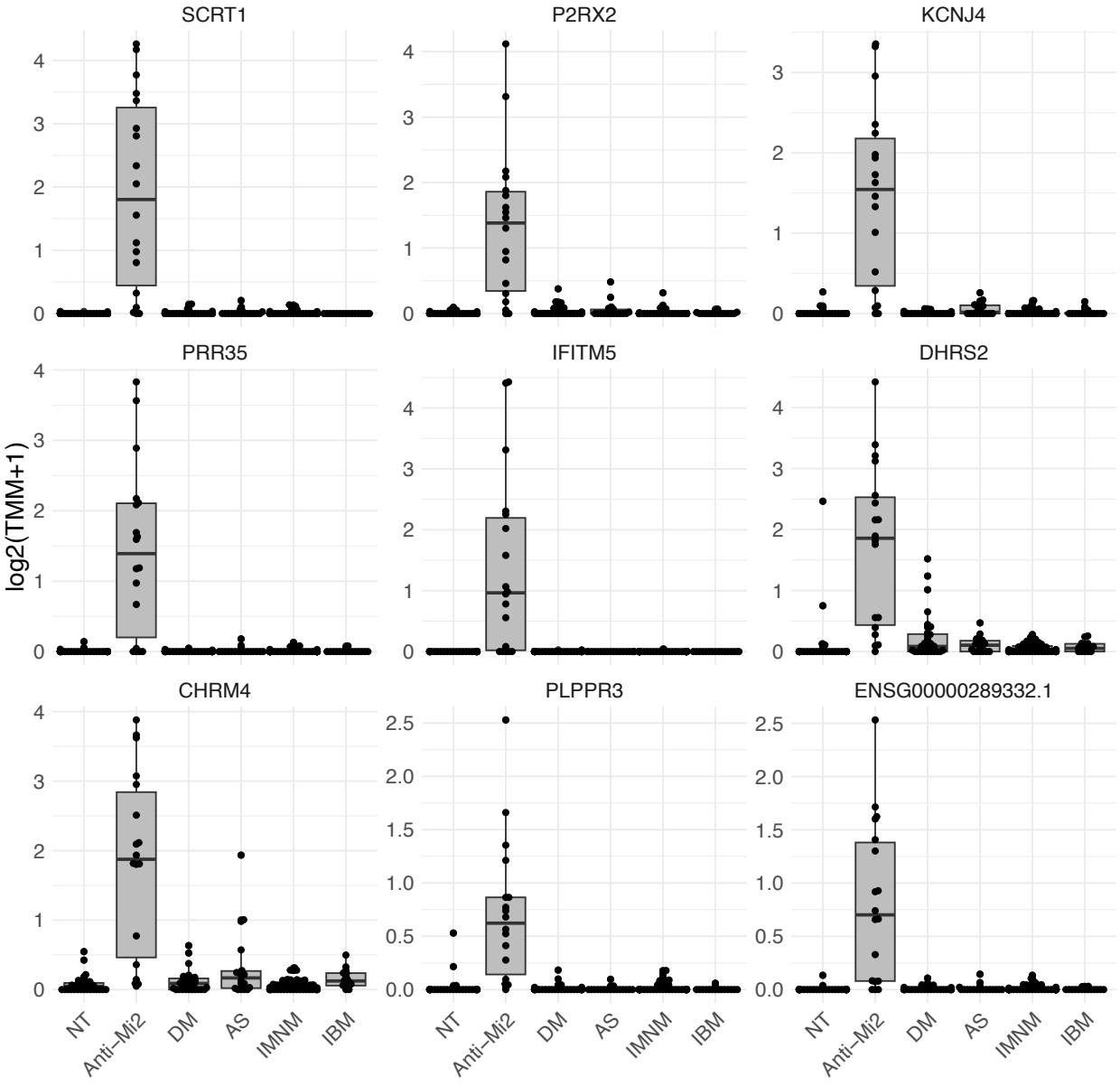
We did note that a number of the anti-Mi2-positive DM patients included in this study did not display the specific transcriptional phenotype observed in the majority of these patients (Figure 3). However, these patients were also those who had the lowest anti-Mi2 autoantibody titers and the least inflammatory muscle biopsies (Figure 5). We

propose that the lack of expression of the transcriptional program associated with anti-Mi2 autoantibodies in these patients simply reflects a very mild phenotype. Alternatively, these patients could constitute a distinct pathogenic subtype within the larger group of anti-Mi2-positive DM patients.

This limitation notwithstanding, we have shown that muscle biopsies from DM patients with anti-Mi2 autoantibodies are characterized by the transcriptional derepression of a heterogeneous set of more than 100 genes, many of which are usually not expressed in skeletal muscle. We demonstrated that this set of genes is expressed in a highly coordinated manner and is highly correlated both with the levels of anti-Mi2 autoantibodies in the blood and transcriptomic markers of disease activity within the muscle. Although anti-Mi2 autoantibodies recognize key components of the NuRD complex, further studies will be required to demonstrate whether the pathogenesis of the disease is mediated by these autoantibodies disrupting the function of this transcriptomic repressor within muscle fibers.

FIGURES AND TABLES

Figure 1. Most differentially overexpressed genes in anti-Mi2 dermatomyositis muscle compared to the rest of the muscle biopsies.



DM: dermatomyositis, AS: antisynthetase syndrome, IBM: inclusion body myositis, IMNM: immune-mediated necrotizing myositis, NT: histologically normal biopsies.

Figure 2. Venn diagram showing the number of genes that were differentially overexpressed (q-value < 0.05) in DM patients with anti-Mi2 autoantibodies compared to other myositis patients and normal muscle biopsies.

DM: dermatomyositis with autoantibodies other than anti-Mi2, AS: antisynthetase syndrome, IBM: inclusion body myositis, IMNM: immune-mediated necrotizing myositis, NT: histologically normal biopsies.

Figure 3. Normalized expression (z-score) of the specifically overexpressed genes in patients with anti-Mi2-positive dermatomyositis (DM) compared with non-Mi2-positive DM.

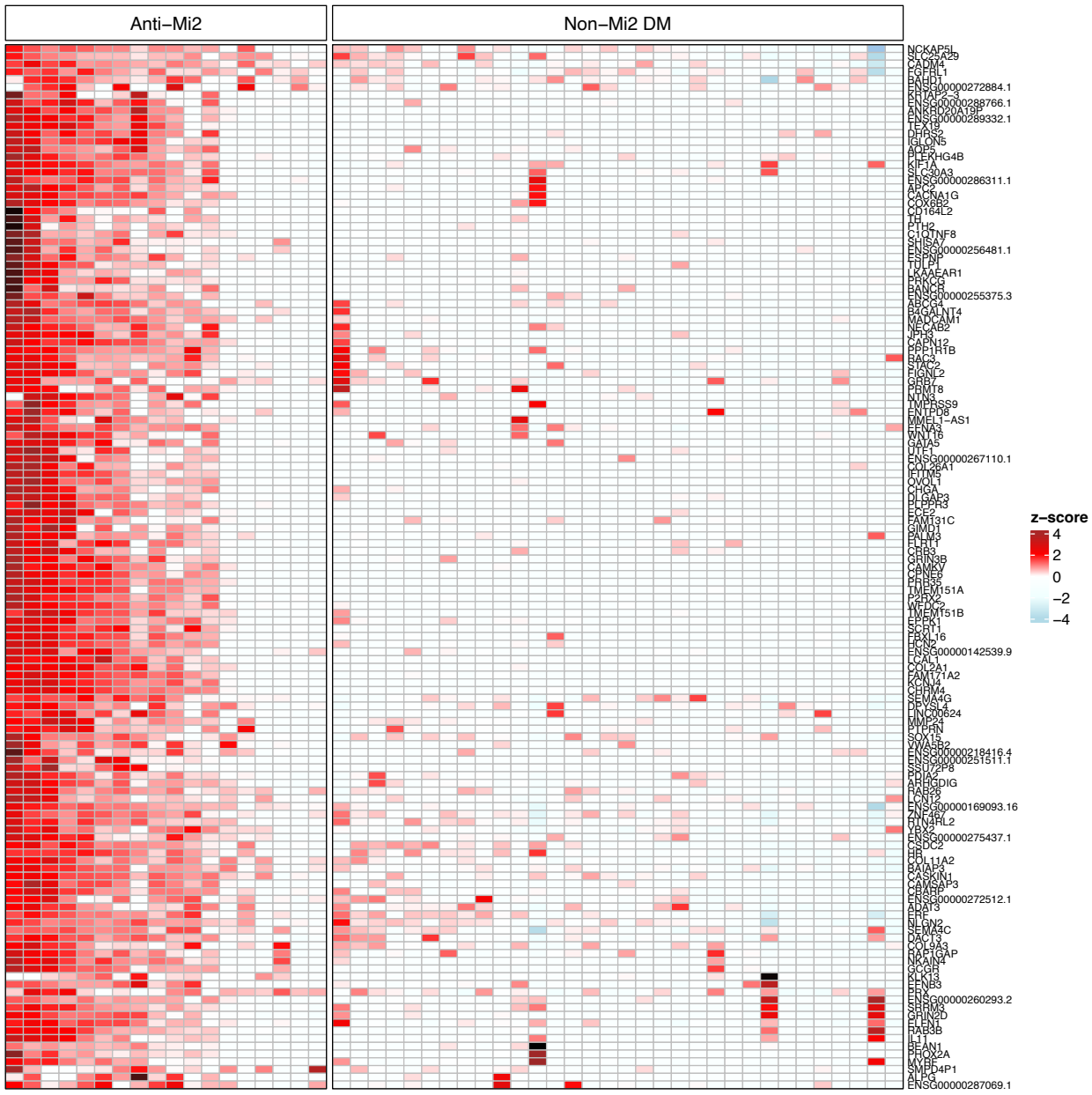


Figure 4. Correlation of expression levels of the specifically overexpressed genes in patients with anti-Mi2-positive dermatomyositis.

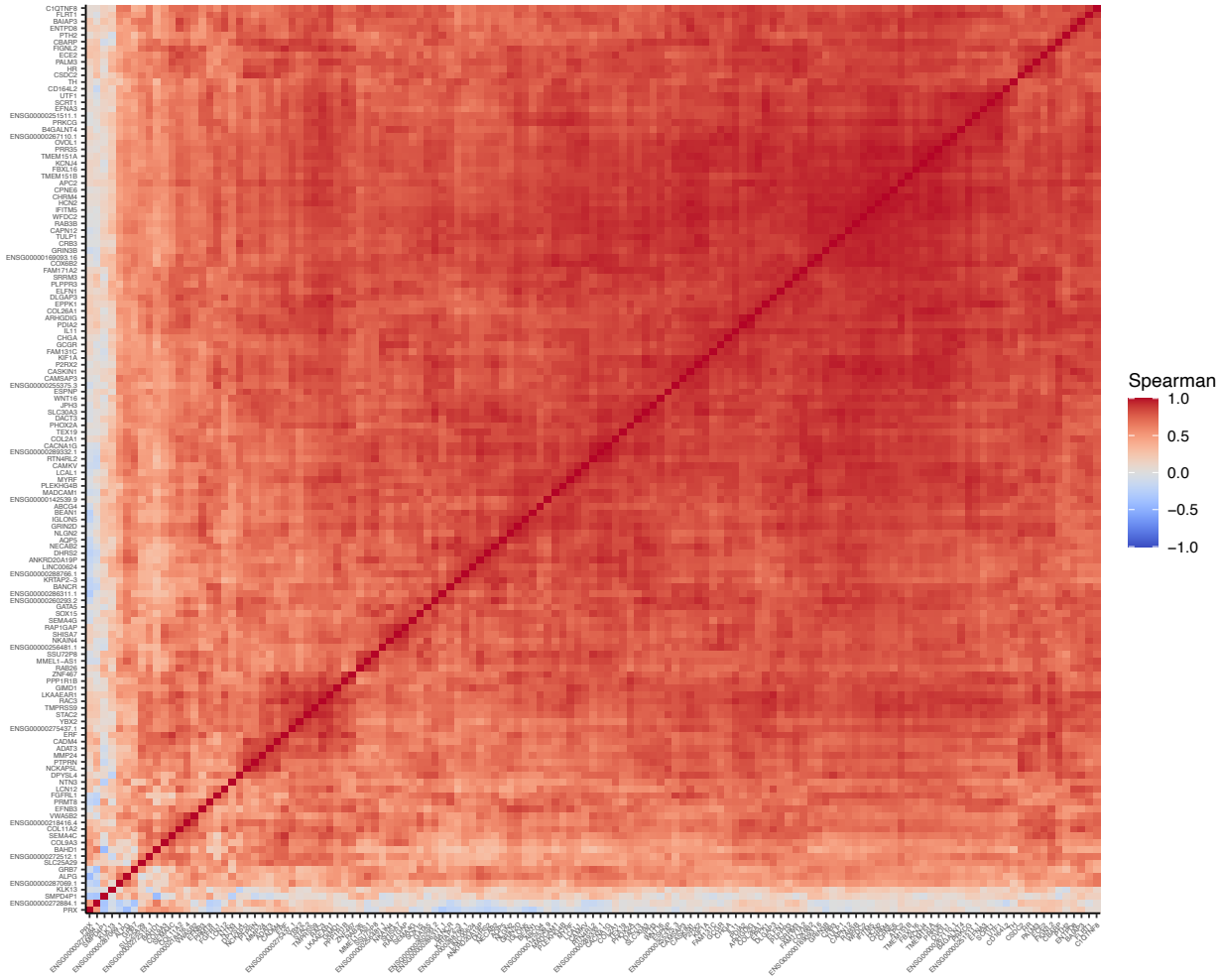
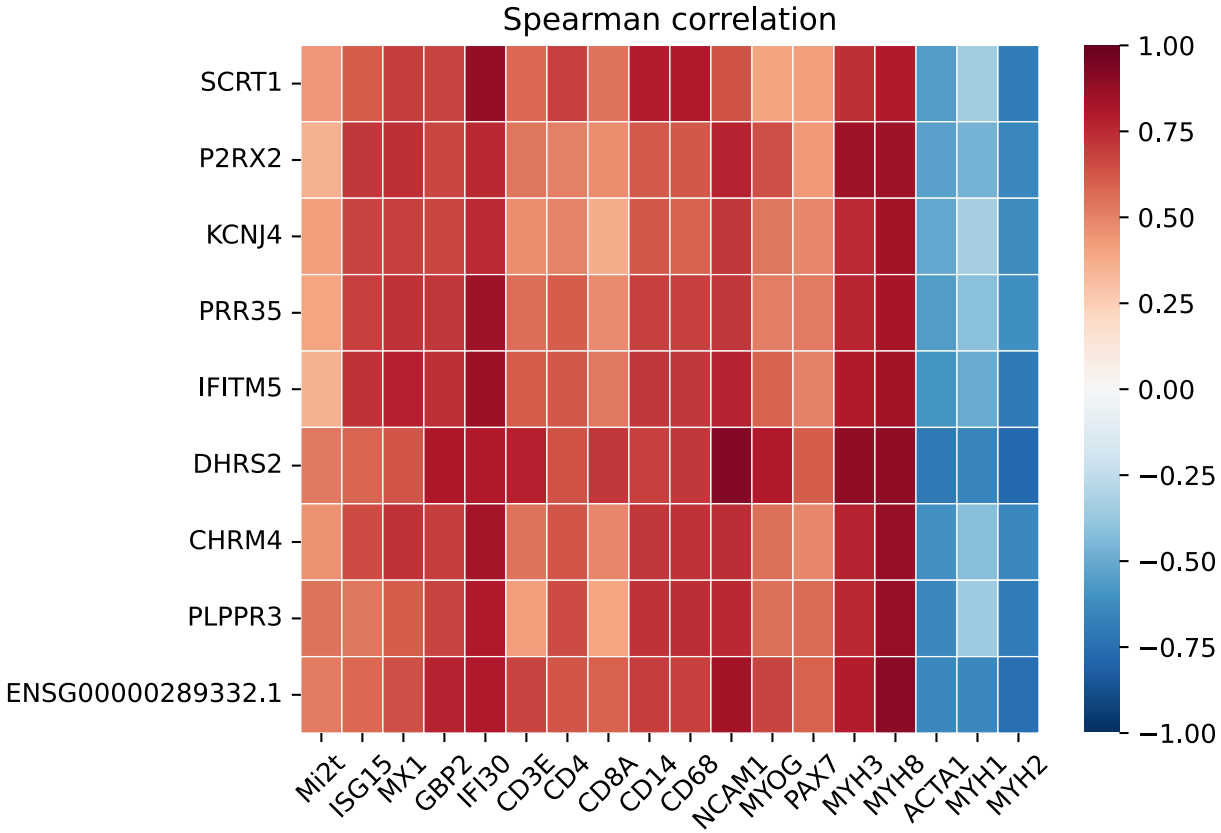


Figure 5. Correlation of the most differentially specifically overexpressed genes in patients with anti-Mi2-positive dermatomyositis and: titer of anti-Mi2 autoantibodies by ELISA (Mi2t), type 1 interferon-inducible genes (ISG15, MX1), type 2 interferon-inducible genes (GBP2, IFI30), T-cell markers (CD3E, CD4, CD8), macrophages (CD14, CD68), markers of muscle differentiation (NCAM1, MYOG, PAX7, MYH3, MYH8), and structural mature muscle proteins (ACTA1, MYH1, MYH2).



Supplementary Table 1. Differential expression of the set of 135 genes specifically overexpressed in anti-Mi2 dermatomyositis.

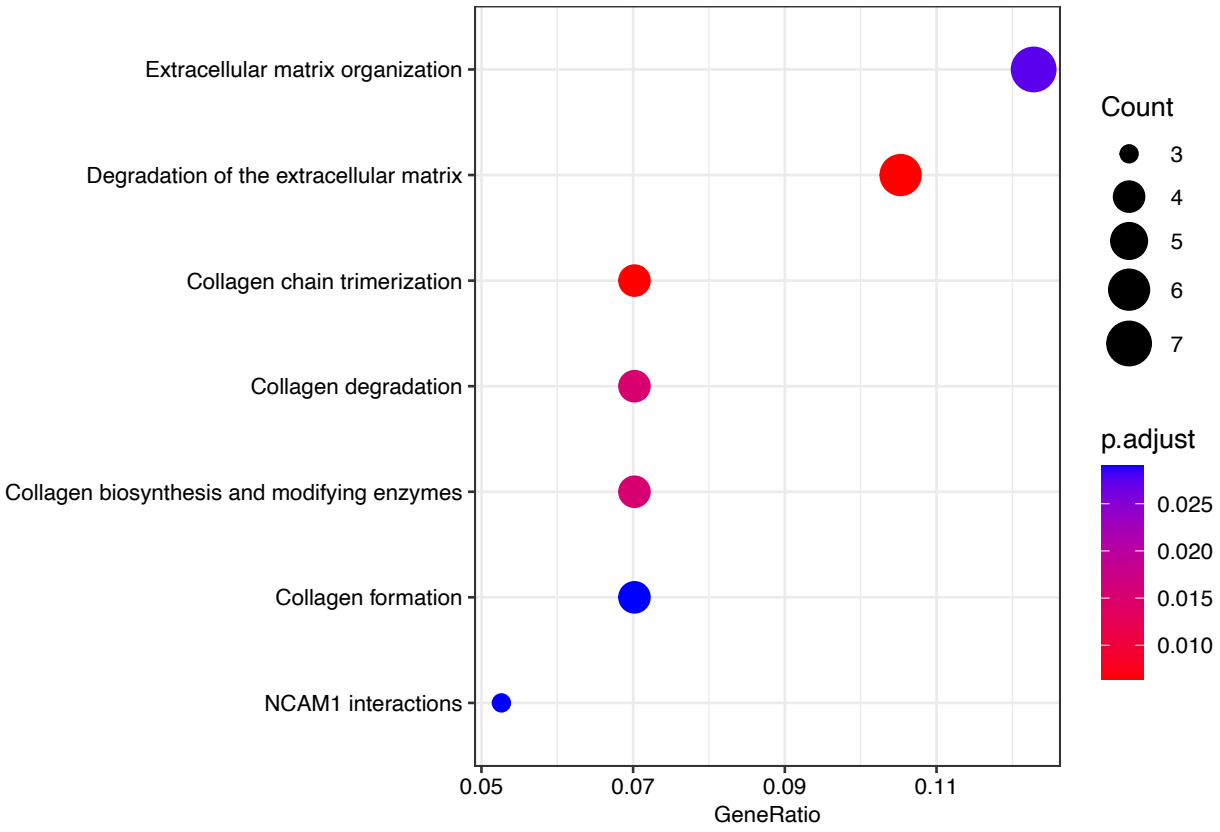
	Anti-Mi2 vs. ALL		Anti-Mi2 vs. NT		Anti-Mi2 vs. DM		Anti-Mi2 vs. AS		Anti-Mi2 vs. IMNM		Anti-Mi2 vs. IBM	
Gene	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**	**logFC**	**adj.P.Val**
ABCG4	2.1	3e-07	2.3	4e-06	2.2	0.002	2.2	0.02	2.0	3e-05	1.5	0.03
ADAT3	1.2	7e-06	1.3	2e-05	1.2	0.04	1.3	0.04	1.2	8e-05	1.2	0.01
ALPG	3.0	3e-07	3.1	4e-06	3.1	0.009	3.1	0.02	2.9	6e-04	3.5	0.004
ANKRD20A19P	4.5	3e-22	4.3	2e-08	4.6	8e-06	4.6	7e-04	4.6	7e-11	4.3	0.002
APC2	2.9	2e-11	3.3	3e-07	2.6	0.009	3.1	0.005	2.8	1e-06	3.3	0.003
AQP5	2.7	2e-09	2.9	7e-09	2.6	0.003	2.8	0.002	2.4	2e-05	2.9	0.001
ARHGDIG	2.7	1e-09	3.4	7e-06	2.4	0.01	3.1	0.01	2.3	4e-05	2.6	0.02
B4GALNT4	3.2	2e-16	3.2	6e-07	2.8	3e-05	3.5	4e-04	3.3	2e-10	3.1	0.001
BAHD1	0.6	2e-06	0.4	0.01	0.7	0.01	0.7	0.007	0.7	7e-06	0.5	0.04
BAIAP3	1.5	2e-13	1.0	9e-05	1.7	3e-04	1.9	3e-04	1.5	5e-10	1.6	0.001
BANCR	2.8	1e-11	2.8	1e-08	2.6	0.003	2.7	0.003	3.0	2e-08	2.6	0.003
BEAN1	2.0	1e-05	3.1	1e-09	2.2	0.02	2.1	0.03	1.3	0.005	1.7	0.03
C1QTNF8	2.9	2e-11	2.7	9e-07	3.0	5e-04	3.2	6e-04	2.9	4e-07	3.1	0.001
CACNA1G	3.1	8e-15	3.4	5e-07	3.2	1e-04	4.0	6e-04	2.7	1e-07	2.9	0.004
CADM4	1.1	5e-10	0.8	4e-04	1.2	3e-04	1.4	2e-04	1.0	5e-08	1.4	2e-04
CAMKV	4.1	7e-21	4.0	1e-09	4.3	3e-07	4.0	3e-04	4.2	1e-12	3.6	0.002
CAMSAP3	3.0	8e-12	3.1	1e-06	2.7	1e-04	2.6	0.004	3.1	3e-08	2.9	0.004
CAPN12	3.0	8e-17	3.8	2e-07	2.8	6e-04	3.6	0.004	2.8	4e-09	2.1	0.03
CASKIN1	2.5	1e-14	2.8	1e-08	3.1	3e-05	2.3	0.003	2.0	3e-08	2.8	6e-04
CBARP	2.3	9e-15	2.8	3e-09	2.0	2e-04	2.1	0.006	2.0	7e-08	3.0	2e-04
CD164L2	1.8	9e-06	1.5	7e-04	2.1	0.005	1.7	0.05	1.7	0.001	1.7	0.03
CHGA	3.4	4e-14	3.4	1e-09	3.5	3e-05	2.8	0.009	3.5	1e-09	4.1	2e-05
CHRM4	5.0	3e-28	5.4	8e-10	4.8	8e-07	4.0	0.01	5.4	3e-14	4.3	0.004
COL11A2	2.0	2e-13	1.2	2e-04	2.4	1e-04	2.4	3e-04	2.0	4e-11	2.4	2e-04
COL26A1	2.7	3e-09	2.5	1e-05	2.6	0.002	2.9	0.004	2.7	1e-06	2.6	0.005
COL2A1	3.8	7e-16	3.7	3e-05	3.7	6e-05	4.0	0.007	3.7	2e-08	3.9	0.007
COL9A3	1.8	8e-08	1.9	2e-04	1.6	0.03	1.8	0.02	2.0	2e-06	1.6	0.03
COX6B2	4.5	2e-19	4.6	7e-09	4.5	1e-04	4.8	6e-04	4.4	3e-10	4.6	0.002
CPNE6	4.4	8e-21	4.4	5e-09	4.3	6e-06	4.7	3e-04	4.5	7e-11	3.7	0.005
CRB3	2.4	4e-08	2.5	4e-06	2.0	0.04	2.6	0.007	2.5	7e-06	2.6	0.003
CSDC2	1.9	1e-06	3.3	4e-09	2.2	0.002	1.7	0.04	1.1	0.005	1.4	0.04
DACT3	1.5	1e-10	2.0	9e-08	1.3	0.01	1.8	0.002	1.4	3e-06	1.4	0.01
DHRS2	5.1	2e-28	6.0	7e-12	4.1	3e-05	4.7	0.001	5.4	1e-14	5.2	6e-04
DLGAP3	2.4	3e-09	2.9	3e-07	2.3	0.002	2.3	0.02	2.2	9e-06	2.2	0.01
DPYSL4	1.4	3e-09	1.4	3e-06	1.3	0.02	1.4	0.02	1.2	3e-05	2.0	6e-04
ECE2	3.9	3e-17	3.6	7e-07	4.0	2e-04	4.0	0.002	4.0	3e-09	4.1	0.002
EFNA3	1.4	4e-07	1.4	2e-04	1.5	0.01	1.5	0.02	1.3	8e-05	1.4	0.02
EFNB3	1.5	1e-04	1.3	0.002	1.9	0.02	1.7	0.03	1.3	0.003	1.7	0.006
ELFN1	1.7	2e-08	1.5	2e-06	2.0	0.004	1.9	0.01	1.5	1e-06	1.6	0.004
ENSG00000142539.9	3.9	1e-13	3.7	6e-07	4.1	5e-05	3.5	0.01	4.2	5e-10	3.8	0.004
ENSG00000169093.16	1.1	2e-11	1.1	8e-07	1.3	8e-04	1.1	0.005	1.0	9e-07	0.9	0.009
ENSG00000218416.4	2.3	5e-07	2.2	2e-06	2.0	0.008	2.0	0.02	2.5	1e-06	2.6	3e-04
ENSG00000251511.1	2.6	8e-13	2.3	2e-07	2.9	6e-05	2.6	0.002	2.6	5e-08	2.7	0.001
ENSG00000255375.3	3.1	2e-11	3.1	2e-07	2.9	0.003	2.5	0.03	3.4	1e-08	3.5	5e-04
ENSG00000256481.1	2.3	3e-10	2.2	2e-08	2.6	7e-05	2.1	0.005	2.3	3e-06	2.5	7e-04
ENSG00000260293.2	2.1	6e-08	2.7	1e-07	1.9	0.04	2.4	0.009	1.9	6e-06	1.9	0.007
ENSG00000267110.1	2.3	2e-05	1.7	0.006	2.5	0.007	2.5	0.01	2.3	3e-04	3.0	7e-04
ENSG00000272512.1	2.9	8e-13	2.5	3e-06	1.7	0.02	3.1	7e-04	3.6	6e-13	3.3	2e-04
ENSG00000272884.1	2.1	6e-04	1.5	0.05	2.3	0.02	3.1	0.004	2.1	4e-04	1.8	0.05
ENSG00000275437.1	2.2	1e-09	2.2	5e-06	2.1	0.002	1.9	0.007	2.4	2e-08	2.1	0.001
ENSG00000286311.1	3.3	2e-14	3.2	4e-10	3.3	2e-05	2.9	4e-04	3.3	4e-11	3.6	2e-05
ENSG00000287069.1	1.7	5e-04	1.4	0.001	2.0	0.02	1.7	0.03	1.7	0.003	1.5	0.05
ENSG00000288766.1	1.8	8e-07	2.3	7e-07	2.2	3e-04	1.9	0.01	1.3	0.002	1.2	0.04
ENSG00000289332.1	4.3	1e-24	4.3	4e-10	4.4	1e-06	4.4	3e-04	4.2	1e-11	4.7	1e-04
ENTPD8	3.2	5e-13	3.5	4e-09	2.5	0.006	3.7	3e-04	3.3	2e-08	3.6	2e-04
EPPK1	2.7	2e-09	3.0	9e-05	2.9	8e-04	3.0	0.006	2.5	9e-07	2.4	0.02
ERF	1.3	2e-08	1.5	3e-07	1.4	0.004	1.1	0.03	1.1	1e-05	1.1	0.02
ESPNP	3.0	1e-11	3.2	9e-09	3.2	4e-04	2.4	0.02	3.1	4e-08	2.9	0.002
FAM131C	2.6	7e-08	3.1	6e-07	2.1	0.03	2.5	0.02	2.6	2e-05	2.3	0.02
FAM171A2	3.1	1e-15	3.7	6e-07	4.2	9e-07	2.6	0.02	2.3	2e-06	3.1	0.005
FBXL16	3.8	8e-17	4.8	1e-08	3.7	3e-05	3.5	0.006	3.7	3e-08	2.9	0.01
FGFRL1	0.9	7e-08	0.4	0.008	1.0	7e-04	1.2	2e-04	0.8	4e-06	1.1	5e-04
FIGNL2	1.8	6e-07	1.8	6e-05	1.9	0.01	1.8	0.01	1.7	2e-05	1.5	0.02
FLRT1	2.5	2e-09	3.0	7e-08	2.1	0.01	3.1	4e-04	2.2	5e-06	2.5	0.004
GATA5	2.3	6e-08	2.4	4e-06	2.2	0.007	2.1	0.02	2.3	1e-05	2.4	0.004
GCGR	4.2	1e-17	4.2	7e-08	4.2	2e-05	4.2	9e-04	4.5	2e-11	3.7	0.006
GIMD1	2.6	1e-12	2.3	3e-07	2.8	1e-04	2.4	0.006	2.6	4e-08	2.9	4e-04
GRB7	1.9	3e-06	2.4	6e-08	1.9	0.03	1.4	0.05	1.8	3e-05	1.8	0.001
GRIN2D	2.7	4e-13	3.9	3e-09	1.8	0.04	2.3	0.02	2.6	8e-08	2.9	0.003
GRIN3B	3.1	2e-12	3.4	5e-08	2.9	5e-04	2.8	0.02	3.3	2e-09	2.8	0.005
HCN2	3.5	1e-15	4.2	3e-07	3.3	5e-04	3.3	0.02	3.5	2e-08	2.9	0.02
HR	1.6	1e-07	2.1	5e-08	1.7	0.01	1.6	0.01	1.3	8e-05	1.3	0.02
IFITM5	5.4	2e-30	5.1	3e-09	5.5	5e-07	5.4	3e-04	5.4	2e-12	5.6	4e-04
IGLON5	3.6	3e-14	3.5	9e-08	3.6	1e-04	3.8	6e-04	3.6	2e-07	3.8	0.001
IL11	3.8	2e-13	4.3	6e-09	3.1	0.01	4.0	0.002	3.8	4e-08	3.9	0.003
JPH3	2.8	4e-11	3.2	6e-08	2.7	5e-04	3.2	6e-04	2.7	4e-08	2.4	0.004
KCNJ4	5.6	6e-34	5.6	5e-11	5.9	6e-09	4.8	0.001	5.7	3e-15	5.8	2e-04
KIF1A	4.7	3e-22	4.8	3e-07	3.7	2e-04	5.3	2e-04	5.5	2e-15	4.0	0.002
KLK13	2.5	4e-06	2.3	1e-05	2.6	0.01	2.6	0.004	2.6	1e-06	2.3	0.007
KRTAP2-3	1.9	4e-06	1.4	0.006	2.3	0.004	1.9	0.05	2.0	1e-04	2.1	0.02
LCAL1	2.0	4e-08	1.7	2e-05	2.4	4e-04	2.0	0.004	2.0	2e-05	2.0	0.004
LCN12	2.0	3e-11	2.2	2e-07	1.7	0.01	2.2	0.005	2.0	2e-08	2.5	7e-04
LINC00624	2.8	1e-11	2.7	8e-08	2.5	6e-04	3.0	3e-04	2.9	1e-08	3.4	3e-05
LKAAEAR1	3.0	6e-14	2.9	2e-08	3.5	3e-06	2.7	0.003	3.0	5e-09	2.9	7e-04
MADCAM1	2.5	4e-20	2.6	1e-07	2.6	5e-05	2.8	0.001	2.3	3e-10	2.7	0.001
MMEL1-AS1	2.0	2e-07	1.7	2e-05	2.2	0.007	2.0	0.007	2.0	1e-05	2.1	0.003
MMP24	1.8	4e-06	2.8	4e-07	2.1	0.004	2.1	0.01	1.1	0.005	1.4	0.04
MYRF	2.3	5e-09	2.9	3e-08	2.2	0.01	2.3	0.006	2.2	7e-09	1.5	0.02
NCKAP5L	0.8	2e-05	1.0	4e-07	0.8	0.005	0.9	0.02	0.4	0.03	0.8	0.007
NECAB2	2.8	1e-10	3.3	4e-08	2.4	0.006	3.1	0.01	2.8	7e-07	2.5	0.01
NKAIN4	4.2	2e-20	4.2	2e-09	4.2	5e-06	4.4	2e-04	4.5	2e-14	3.7	0.002
NLGN2	1.0	1e-12	1.3	1e-11	0.8	0.007	1.0	7e-04	0.9	2e-08	1.0	4e-04
NTN3	1.4	9e-04	0.9	0.04	1.6	0.03	1.6	0.02	1.5	0.002	1.4	0.04
OVOL1	3.1	7e-15	2.8	9e-08	3.4	1e-05	3.2	4e-04	3.2	8e-10	3.0	0.002
P2RX2	5.5	8e-35	5.5	2e-12	5.1	9e-08	5.0	3e-04	5.8	1e-17	5.8	2e-05
PALM3	3.0	4e-12	3.1	7e-08	3.3	2e-04	2.9	0.007	3.0	6e-08	2.7	0.005
PDIA2	2.5	8e-13	3.1	7e-08	2.0	0.009	2.8	0.002	2.5	4e-08	2.5	0.003
PHOX2A	2.2	8e-08	2.1	4e-06	2.3	0.009	2.4	0.002	2.2	1e-05	2.7	4e-04
PLEKHG4B	2.2	6e-06	1.7	0.002	2.3	0.004	2.5	0.009	2.5	1e-06	1.7	0.02
PLPPR3	4.3	1e-25	4.2	3e-11	4.5	1e-08	4.5	9e-06	4.2	1e-13	4.8	5e-06
PPP1R1B	3.9	2e-14	4.4	1e-07	3.5	4e-04	4.6	9e-04	3.9	6e-09	2.6	0.02
PRKCG	3.5	4e-14	3.4	2e-07	3.5	3e-04	3.6	0.002	3.7	6e-10	2.7	0.02
PRMT8	2.2	9e-06	2.2	1e-05	1.9	0.05	2.4	0.006	2.3	1e-05	1.7	0.04
PRR35	5.5	1e-31	5.4	6e-10	5.7	5e-08	5.4	4e-04	5.5	2e-13	5.7	3e-04
PRX	1.1	3e-07	0.8	0.01	1.1	0.01	1.3	0.003	1.2	5e-07	1.0	0.01
PTH2	1.7	9e-06	1.4	5e-04	1.8	0.02	1.7	0.04	1.7	7e-04	2.0	0.006
PTPRN	3.5	6e-13	4.1	2e-09	2.4	0.04	3.4	0.007	3.8	2e-09	3.6	0.002
RAB26	1.9	6e-08	1.0	0.006	2.4	7e-04	2.6	7e-04	1.9	9e-08	1.7	0.005
RAB3B	4.0	1e-16	3.6	5e-07	3.9	1e-04	3.9	4e-04	4.7	1e-15	3.5	0.002
RAC3	2.3	8e-09	2.8	2e-06	1.9	0.007	2.1	0.01	2.4	3e-08	2.0	0.01
RAP1GAP	2.4	5e-09	2.8	4e-06	2.0	0.02	2.5	0.006	2.7	2e-08	1.8	0.02
RTN4RL2	1.2	4e-08	1.2	2e-05	1.1	0.05	1.3	0.02	1.1	7e-06	1.4	0.002
SCRT1	6.3	2e-39	6.3	2e-11	6.3	1e-08	6.0	3e-04	6.3	4e-15	6.9	9e-05
SEMA4C	0.7	5e-06	0.7	8e-05	0.8	0.004	0.8	0.004	0.5	0.002	0.6	0.005
SEMA4G	0.9	9e-04	0.6	0.008	1.6	0.008	1.0	0.03	0.6	0.009	0.8	0.04
SHISA7	3.4	3e-14	3.0	4e-07	3.6	2e-05	3.6	4e-04	3.7	1e-11	3.2	0.002
SLC25A29	0.8	6e-07	0.7	4e-05	0.7	0.04	0.9	0.007	0.8	2e-07	0.7	0.002
SLC30A3	2.9	1e-11	3.8	2e-07	3.1	0.002	2.5	0.04	2.6	7e-07	2.6	0.02
SMPD4P1	2.1	6e-05	1.5	0.004	2.0	0.01	2.3	0.005	2.2	1e-04	3.0	9e-05
SOX15	1.3	1e-04	1.0	0.002	1.8	0.02	1.9	0.01	0.9	9e-04	1.2	0.02
SRRM3	2.2	1e-11	2.8	5e-08	1.8	0.03	1.9	0.02	2.1	6e-07	2.4	0.004
SSU72P8	2.0	1e-07	1.7	9e-05	2.3	0.003	2.0	0.02	2.0	1e-04	2.3	0.004
STAC2	2.4	4e-09	2.8	1e-06	2.3	0.008	2.2	0.02	2.1	4e-05	2.7	0.003
TEX19	3.3	2e-13	4.0	3e-11	3.5	1e-05	2.6	0.02	3.1	3e-07	3.3	0.002
TH	3.0	3e-12	2.8	4e-07	3.3	7e-05	2.8	0.004	3.0	5e-08	3.3	3e-04
TMEM151A	4.1	1e-22	4.1	5e-10	4.0	7e-06	3.8	8e-04	4.1	2e-11	4.3	2e-04
TMEM151B	3.6	1e-15	3.3	2e-07	3.4	8e-05	3.8	2e-04	3.8	3e-11	3.6	3e-04
TMPRSS9	3.1	5e-11	2.8	2e-07	3.3	2e-04	2.3	0.02	3.3	9e-11	3.3	3e-04
TULP1	2.7	6e-10	3.2	3e-09	2.7	0.002	2.5	0.02	2.7	2e-06	2.4	0.01
UTF1	2.4	3e-07	2.2	7e-05	2.7	0.003	2.6	0.008	2.4	8e-05	2.1	0.03
VWA5B2	2.3	8e-08	2.3	5e-06	2.7	3e-05	2.7	2e-04	1.9	6e-05	2.5	4e-04
WFDC2	4.2	1e-24	5.1	2e-09	4.4	5e-07	4.1	0.003	3.6	3e-09	4.5	0.001
WNT16	2.9	8e-10	3.0	3e-08	2.4	0.02	3.0	0.004	3.0	7e-07	3.3	8e-04
YBX2	1.6	3e-13	2.0	2e-08	1.5	8e-04	1.6	0.004	1.6	5e-09	1.4	0.01
ZNF467	1.2	3e-08	1.3	4e-06	1.3	0.006	1.3	0.007	1.0	1e-04	1.1	0.01

ALL: all the other muscle biopsies, NT: histologically normal biopsies, DM: dermatomyositis, AS: antisynthetase syndrome, IBM: inclusion body myositis, IMNM: immune-mediated necrotizing myositis.

Supplementary Figure 1. Venn diagram showing the number of genes that were differentially underexpressed (q-value < 0.05) in DM patients with anti-Mi2 autoantibodies compared to other myositis patients and normal muscle biopsies.

DM: dermatomyositis, AS: antisynthetase syndrome, IBM: inclusion body myositis, IMNM: immune-mediated necrotizing myositis, NT: histologically normal biopsies.

Supplementary Figure 2. Pathway enrichment analysis of the set of 126 genes specifically overexpressed in anti-Mi2 using the Reactome database.



REFERENCES

1. Casal-Dominguez M, Pinal-Fernandez I, Pak K, Huang W, Selva-O'Callaghan A, Albayda J, et al. Performance of the 2017 European Alliance of Associations for Rheumatology/American College of Rheumatology Classification Criteria for Idiopathic Inflammatory Myopathies in Patients With Myositis-Specific Autoantibodies. Arthritis Rheumatol. 2022 Mar; 74(3):508-517.

2. Allenbach Y, Arouche-Delaperche L, Preusse C, Radbruch H, Butler-Browne G, Champtiaux N, et al. Necrosis in anti-SRP(+) and anti-HMGCR(+)myopathies: Role of autoantibodies and complement. Neurology. 2018 Jan 12.

3. Arouche-Delaperche L, Allenbach Y, Amelin D, Preusse C, Mouly V, Mauhin W, et al. Pathogenic role of anti-signal recognition protein and anti-3-Hydroxy-3-methylglutaryl-CoA reductase antibodies in necrotizing myopathies: Myofiber atrophy and impairment of muscle regeneration in necrotizing autoimmune myopathies. Ann Neurol. 2017 Apr; 81(4):538-548.

4. Bergua C, Chiavelli H, Allenbach Y, Arouche-Delaperche L, Arnoult C, Bourdenet G, et al. In vivo pathogenicity of IgG from patients with anti-SRP or anti-HMGCR autoantibodies in immune-mediated necrotising myopathy. Ann Rheum Dis. 2019 Jan; 78(1):131-139.

5. Pinal-Fernandez I, Mecoli CA, Casal-Dominguez M, Pak K, Hosono Y, Huapaya J, et al. More prominent muscle involvement in patients with dermatomyositis with anti-Mi2 autoantibodies. Neurology. 2019 Nov 5; 93(19):e1768-e1777.

6. Pinal-Fernandez I, Pak K, Casal-Dominguez M, Hosono Y, Mecoli C, Christopher-Stine L, et al. Validation of anti-Mi2 autoantibody testing by line blot. Autoimmun Rev. 2020 Jan; 19(1):102425.

7. Tanboon J, Inoue M, Hirakawa S, Tachimori H, Hayashi S, Noguchi S, et al. Pathologic Features of Anti-Mi-2 Dermatomyositis. Neurology. 2021 Jan 19; 96(3):e448-e459.

8. Fornaro M, Girolamo F, Cavagna L, Franceschini F, Giannini M, Amati A, et al. Severe muscle damage with myofiber necrosis and macrophage infiltrates characterize anti-Mi2 positive dermatomyositis. Rheumatology (Oxford). 2021 Jun 18; 60(6):2916-2926.

9. Seelig HP, Moosbrugger I, Ehrfeld H, Fink T, Renz M, Genth E. The major dermatomyositis-specific Mi-2 autoantigen is a presumed helicase involved in transcriptional activation. Arthritis Rheum. 1995 Oct; 38(10):1389-1399.

10. Pinal-Fernandez I, Casal-Dominguez M, Derfoul A, Pak K, Miller FW, Milisenda JC, et al. Machine learning algorithms reveal unique gene expression profiles in muscle biopsies from patients with different types of myositis. Ann Rheum Dis. 2020 Sep; 79(9):1234-1242.

11. Lloyd TE, Mammen AL, Amato AA, Weiss MD, Needham M, Greenberg SA. Evaluation and construction of diagnostic criteria for inclusion body myositis. Neurology. 2014 Jul 29; 83(5):426-433.

12. Pinal-Fernandez I, Casal-Dominguez M, Derfoul A, Pak K, Plotz P, Miller FW, et al. Identification of distinctive interferon gene signatures in different types of myositis. Neurology. 2019 Sep 17; 93(12):e1193-e1204.

13. Pinal-Fernandez I, Amici DR, Parks CA, Derfoul A, Casal-Dominguez M, Pak K, et al. Myositis Autoantigen Expression Correlates With Muscle Regeneration but Not Autoantibody Specificity. Arthritis Rheumatol. 2019 Aug; 71(8):1371-1376.

14. Amici DR, Pinal-Fernandez I, Mazala DA, Lloyd TE, Corse AM, Christopher-Stine L, et al. Calcium dysregulation, functional calpainopathy, and endoplasmic reticulum stress in sporadic inclusion body myositis. Acta Neuropathol Commun. 2017 Mar 22; 5(1):24.

15. Amici DR, Pinal-Fernandez I, Christopher-Stine L, Mammen AL, Mendillo ML. A network of core and subtype-specific gene expression programs in myositis. Acta Neuropathol. 2021 Nov; 142(5):887-898.