

PEC 1: MICROARRAY ANALYSIS OF GEO:GSE1659 "Time series of diabetes and exercise training induced expression changes in skeletal muscle of mice"

Github: <https://github.com/sofiasofia2208/sofiazdrad>

TABLA DE CONTENIDOS

<i>Abstract</i>	<i>p.1</i>
<i>1. Objetivos</i>	<i>p.2</i>
<i>2. Material y métodos</i>	<i>p.2</i>
<i>2.1 Diseño experimental</i>	<i>p.3</i>
<i>3. Resultados</i>	<i>p.3</i>
<i>3.1. PASO 1: Preparación de los datos</i>	<i>p.3</i>
<i>3.2. PASO 2: Control de calidad</i>	<i>p.4</i>
<i>3.3. PASO 3: Normalización y control de calidad de datos normalizados</i>	<i>p.7</i>
<i>3.4. PASO 4: Detección de efectos derivados del "batch"</i>	<i>p.8</i>
<i>3.5. PASO 5: Detección de los genes más y menos variables y filtraje</i>	<i>p.9</i>
<i>3.6. PASO 6: Matriz de diseño y matriz de contraste</i>	<i>p.10</i>
<i>3.7. PASO 7: selección de genes diferencialmente expresados</i>	<i>p.11</i>
<i>3.7.1. Volcano plots</i>	<i>p.12</i>
<i>3.7.2. Comparaciones múltiples y diagrama de Venn</i>	<i>p.15</i>
<i>3.7.3. Visualización de los perfiles de expresión usando "Heatmaps"</i>	<i>p.15</i>
<i>3.8. PASO 8: Análisis de significación biológica</i>	<i>p.17</i>
<i>4. Discusión</i>	<i>p.21</i>
<i>5. Conclusión</i>	<i>p.22</i>
<i>6. Bibliografía</i>	<i>p.22</i>
<i>Anexo: Código R utilizado</i>	<i>p.23</i>

Abstract

Se ha evaluado la influencia del ejercicio en ratones sanos y en ratones diabéticos. El análisis de las muestras de microarray reveló una disminución de genes implicados en la contracción muscular cuando el ratón sano es entrenado, de genes implicados en la síntesis proteica entre ratones diabéticos y control, así como diferencias en genes implicados en la síntesis de colágeno cuando se compararon ratones diabéticos que habían hecho ejercicio con el control.

1. Objetivos

El objetivo de este trabajo es doble. Por un lado, se estudiará si existen genes diferencialmente expresados en el músculo esquelético de ratones con y sin diabetes a través de muestras de microarray. Por otro lado, se estudiará si la realización de actividad física conlleva una expresión diferencial de genes, tanto en ratones diabéticos como control.

Si nos fijamos en los artículos ya publicados usándose el dataset elegido (Lehti et al., 2006, 2007; Kivelä et al. 2006), vemos que los objetivos del presente estudio no son los mismos, y los grupos que han sido utilizados aquí por tanto difieren del número usado por los autores. Esta cuestión será explicada más extensamente en el apartado de materiales y métodos.

2. Material y métodos

Los datos utilizados en este trabajo proceden del dataset **"GSE1659"** del repositorio Gene Expression Omnibus (GEO), y pertenecen al estudio **"Time series of diabetes and exercise training induced expression changes in skeletal muscle of mice"** realizado por Silvennoinen y colaboradores. El conjunto de datos de expresión de microarray puede descargarse aquí: <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE1659>

El experimento se realizó en ratones machos NMRI de 10 a 15 semanas de edad alojados en condiciones estándar y respetando el bienestar animal. Los animales fueron divididos al azar en grupos de sanos y diabéticos (diabetes inducida por estreptozotocina). En el trabajo original existían 12 grupos, tal y como mencionan en el primer artículo (Lehti et al. 2006): *"Diabetic and healthy animals were randomly assigned into 12 groups (n = 5 per group), which were either sedentary or trained for 1, 3 or 5 weeks. Groups were named as follows: sedentary healthy mice (C1, C3, C5), trained healthy mice (T1, T3, T5), sedentary diabetic mice (D1, D3, D5) and trained diabetic mice (DT1, DT3, DT5)"*.

Las variables que componían el estudio original fueron:

- **Estado de salud "Health status"**, con dos niveles: "Healthy" y "Diabetic"
- **Entrenamiento "Trained"**, con dos niveles "No trained" y "Trained"
- **Tiempo de entrenamiento en semanas "Time"** con tres niveles: 1, 3 o 5 semanas

Sin embargo, en este estudio se ha decidido no utilizar la variable "Tiempo de entrenamiento en semanas (Time)" y sí las variables "Estado de salud (Health status)" y "Entrenamiento (Trained)". Con ambas, se han agrupado las muestras de microarray en 1 grupo (factor) con cuatro niveles combinación de las distintas condiciones experimentales:

- Control sin entrenamiento (Healthy) = 3 muestras [C1, C3 y C5]
- Control con entrenamiento (Healthy Trained) = 3 muestras [T1, T3 y T5]
- Diabético sin entrenamiento (Diabetic) = 3 muestras [D1, D3 y D5]
- Diabético con entrenamiento (Diabetic Trained) = 3 muestras [DT1, DT3 y DT5]

Nota: Para realizar la interpretación biológica de los resultados, sí se ha mantenido en la etiqueta el nº de semanas de ejercicio en todas las muestras. Así, puede llegar a ser posible asociar algún comportamiento diferente de alguna muestra con este hecho.

Por tanto, este estudio está conformado por n = 12 muestras de microarray que han sido obtenidas cada una de un batch de RNA procedente de 5 ratones.

2.1. Diseño experimental

Es un experimento de tipo comparativo.

Obtención de RNA: Tras realizar la eutanasia a los ratones 24 horas después de su último entrenamiento (en caso de que tuvieran), los músculos de la pantorrilla fueron removidos, disecados sin grasa ni tejido conectivo, pesados, congelados en nitrógeno líquido y almacenados a -80°C para su posterior análisis.

Las muestras de RNA se analizaron con el chip *Affymetrix Gene Chip MG U74Av2* (Affymetrix, Inc., Santa Clara, CA). Este chip fue escaneado con *GeneArray Scanner G2500A* (Agilent, Palo Alto, CA). Las imágenes obtenidas en el array fueron analizadas con *Microarray Suite 5.0* (Affymetrix) software.

Los datos fueron analizados usando el software estadístico R, así como librerías de Bioconductor para el análisis de muestras de microarray.

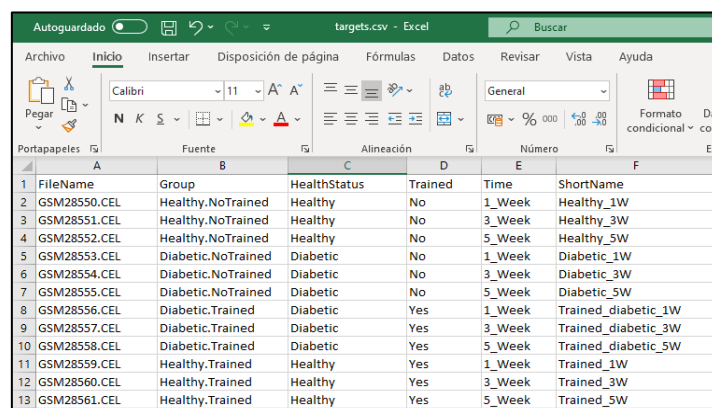
Los pasos realizados en el análisis fueron los siguientes: (1) Preparación de los datos, (2) Medición de calidad de las muestras, (3) Normalización de datos con el método RMA y control de calidad de dichos datos, (4) Detección de efectos derivados del "batch", (5) Detección de genes más variables y filtrado, (6) Matriz de diseño y de contraste, (7) Selección de genes diferencialmente expresados, (8) Significación biológica. Se siguieron las instrucciones de dos documentos proporcionados en esta asignatura: Apartados 6 a 8 del Módulo 2 y Case_Study_1-Microarrays_Analysis.html. Los paquetes requeridos tanto de R como de Bioconductor son indicados en este último documento.

A continuación, en el apartado "Resultados", se ha procurado explicar cada uno de los pasos realizados. Aunque dicha parte corresponda a la metodología, se ha considerado de mayor utilidad explicar cómo se ha realizado cada uno de los pasos en el momento de realizarlos, además de presentar los resultados.

3. Resultados

3.1. PASO 1: Preparación de los datos

En primer lugar hemos definido los directorios donde vamos a almacenar tanto los datos como los resultados de este trabajo. Previamente habíamos creado un documento .csv llamado "targets.csv" con la información relativa a cada muestra tal y como se muestra en la Figura 1.



File Name	Group	Health Status	Trained	Time	Short Name
GSM28550.CEL	Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
GSM28551.CEL	Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
GSM28552.CEL	Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
GSM28553.CEL	Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
GSM28554.CEL	Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
GSM28555.CEL	Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
GSM28556.CEL	Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
GSM28557.CEL	Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
GSM28558.CEL	Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W
GSM28559.CEL	Healthy.Trained	Healthy	Yes	1_Week	Trained_1W
GSM28560.CEL	Healthy.Trained	Healthy	Yes	3_Week	Trained_3W
GSM28561.CEL	Healthy.Trained	Healthy	Yes	5_Week	Trained_5W

Figura 1. Datos fichero targets.csv

Una vez tenemos nuestros directorios "data" y "results", cargado en R el archivo targets.csv, podemos visualizar el conjunto de datos de microarray en una tabla (Tabla 1).

Tabla 1. Datos fichero targets.csv una vez cargados en R

i.FileName	Group	HealthStatus	Trained	Time	ShortName
GSM28550.CEL	Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
GSM28551.CEL	Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
GSM28552.CEL	Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
GSM28553.CEL	Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
GSM28554.CEL	Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
GSM28555.CEL	Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
GSM28556.CEL	Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
GSM28557.CEL	Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
GSM28558.CEL	Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W
GSM28559.CEL	Healthy.Trained	Healthy	Yes	1_Week	Trained_1W
GSM28560.CEL	Healthy.Trained	Healthy	Yes	3_Week	Trained_3W
GSM28561.CEL	Healthy.Trained	Healthy	Yes	5_Week	Trained_5W

A continuación, cargamos los ficheros .CEL y leemos los datos. Al final de este paso tendremos las intensidades "raw" guardadas en "rawData".

Figura 2. Ficheros .CEL asociados a cada una de las muestras de microarray usadas en este estudio.

	Group <chr>	HealthStatus <chr>	Trained <chr>	Time <chr>	ShortName <chr>
GSM28550.CEL	Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
GSM28551.CEL	Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
GSM28552.CEL	Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
GSM28553.CEL	Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
GSM28554.CEL	Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
GSM28555.CEL	Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
GSM28556.CEL	Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
GSM28557.CEL	Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
GSM28558.CEL	Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W
GSM28559.CEL	Healthy.Trained	Healthy	Yes	1_Week	Trained_1W

3.2.PASO 2: Control de calidad

En primer lugar, a través de un histograma (Figura 3), podemos ver la distribución de la señal de cada uno de las muestras de microarray. Como vemos, no se diferencian especialmente unas de otras, teniendo una distribución similar todas las muestras. Asimismo, parecen seguir una distribución normal.

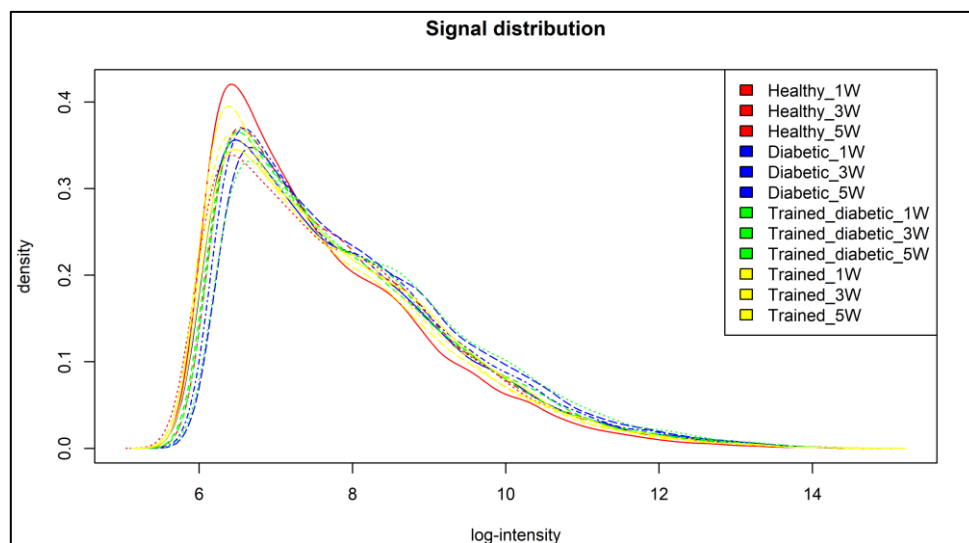


Figura 3. Histograma con las señales de los 12 microarrays estudiados.

También hemos representado los datos crudos en un diagrama de cajas y bigotes (Figura 4). Como vemos, todas las muestras se distribuyen de manera bastante similar.

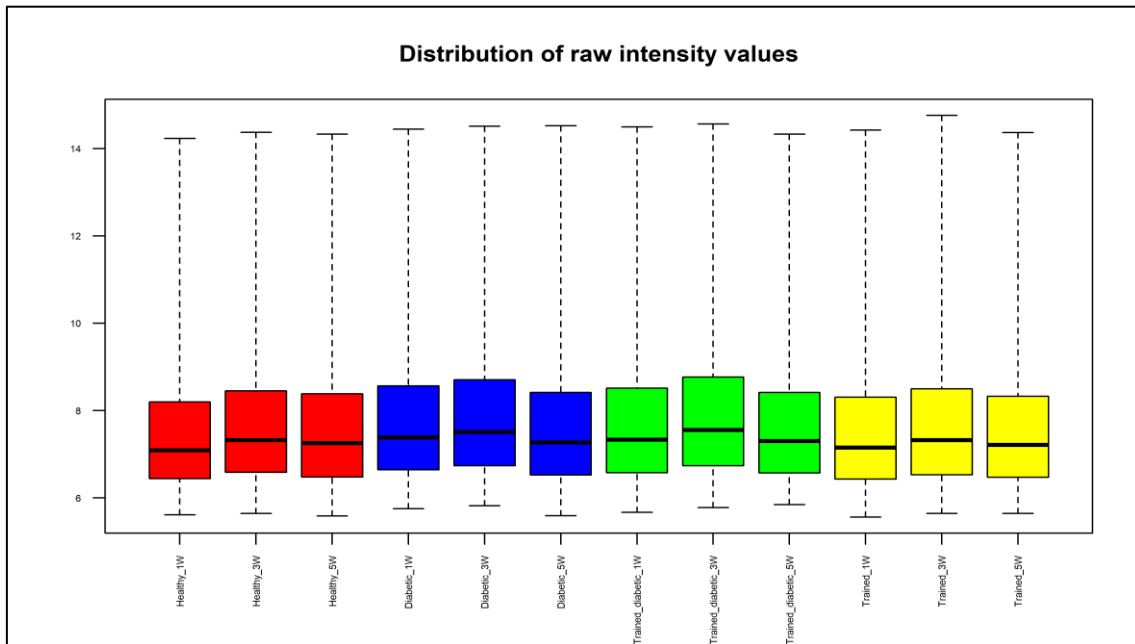


Figura 4. Diagrama de cajas y bigotes correspondientes a los 12 microarrays estudiados.

Asimismo, hemos realizado un Análisis de Componentes Principales (PCA) y representado gráficamente (Figura 5). Podemos observar que el primer componente, PC1, acumula el 33,5% de la varianza, mientras que el PC2 acumula el 29,2%. Aunque estos valores son relativamente bajos, podemos ver cómo en el eje X (PC1), las muestras de ratones de 5 semanas (con y sin entrenamiento), se agrupan en sentido positivo del eje. Pero tampoco encontramos agrupaciones claras ni ninguna muestra que se distribuya muy separada de las demás y tenemos que tener en cuenta que el % de la varianza explicada por cada PC es bajo.

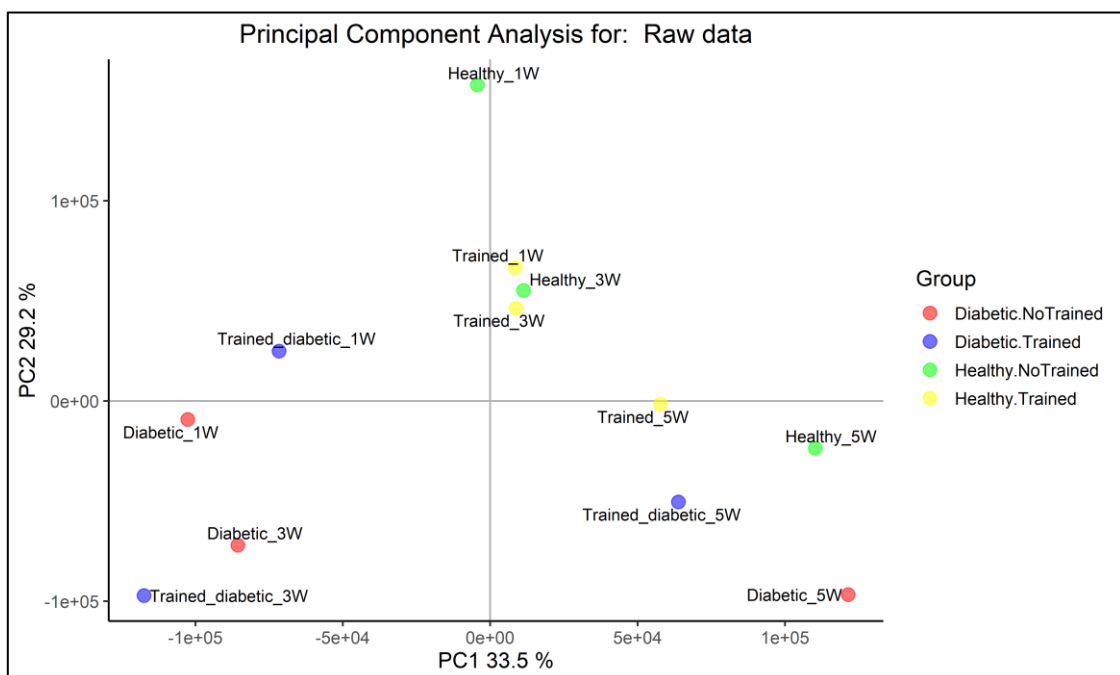


Figura 5. Análisis de componentes principales para las 12 muestras de microarray analizadas.

Asimismo, para ver si las muestras se agrupan por condiciones experimentales, se llevó a cabo un clúster jerárquico (Figura 6). Podemos ver que las muestras de ratones diabéticos se agrupan por un lado, tanto con entrenamiento como sin entrenamiento durante 1 o 3 semanas. Por otro lado, se comportan igual las muestras de ratones sanos, agrupándose por otro lado tanto aquellos ratones con entrenamiento como sin entrenamiento durante 1 o 3 semanas. Resulta curioso que, como ha sido observado en el PCA, todas las muestras de 5 semanas se agrupan juntas.

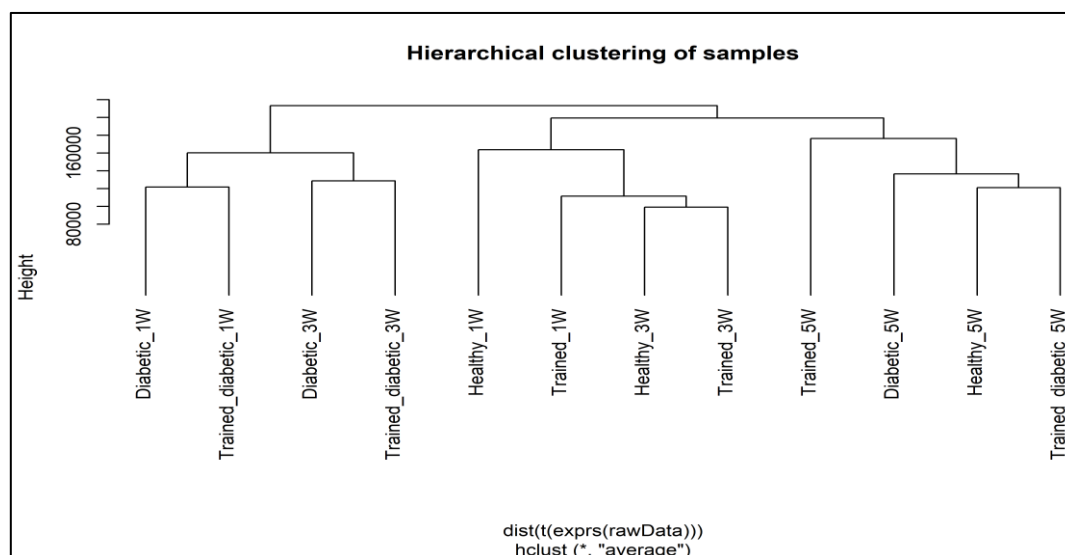


Figura 6. Análisis jerárquico para las 12 muestras de microarray analizadas.

Finalmente, se utilizó la librería "arrayQualityMetrics" para generar un informe de la calidad de nuestras muestras. Este informe llamado "index.html" se ha generado en la carpeta Results/QCDir.Norm.

En la Tabla 2 podemos ver la tabla de muestras de microarray a las que hemos evaluado la calidad por tres métodos. Vemos que solo aparece una X una única vez en una de las muestras, la de Diabetic.NoTrained (Diabetic_5W). Como solo han sido detectado outliers en una de las muestras y solamente por un método, se considera que todas las muestras tienen la suficiente calidad para usarlas para estudio. En el siguiente paso realizaremos la normalización de los datos.

Tabla 2. Análisis de calidad mediante 3 métodos de los datos de microarray crudos "raw". En aquellos test donde se encontró un outlier se representó con una cruz

array	sampleNames	*1	*2	*3	Group	HealthStatus	Trained	Time	ShortName
<input type="checkbox"/> 1	Healthy_1W				Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
<input type="checkbox"/> 2	Healthy_3W				Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
<input type="checkbox"/> 3	Healthy_5W				Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
<input type="checkbox"/> 4	Diabetic_1W				Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
<input type="checkbox"/> 5	Diabetic_3W				Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
<input type="checkbox"/> 6	Diabetic_5W	x			Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
<input type="checkbox"/> 7	Trained_diabetic_1W				Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
<input type="checkbox"/> 8	Trained_diabetic_3W				Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
<input type="checkbox"/> 9	Trained_diabetic_5W				Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W
<input type="checkbox"/> 10	Trained_1W				Healthy.Trained	Healthy	Yes	1_Week	Trained_1W
<input type="checkbox"/> 11	Trained_3W				Healthy.Trained	Healthy	Yes	3_Week	Trained_3W
<input type="checkbox"/> 12	Trained_5W				Healthy.Trained	Healthy	Yes	5_Week	Trained_5W

3.3. PASO 3: Normalización y control de calidad de datos normalizados

El siguiente paso del análisis de nuestros datos fue realizar la normalización de las muestras. Se llevó a cabo un procesamiento mediante el método RMA. Con este paso, homogenizaremos las muestras para una mejor y más fiable comparación.

Se obtuvo de nuevo el informe de calidad pero con nuestras muestras normalizadas. Abrimos el fichero "index.html" que se ha generado y observamos la tabla con el control de calidad mediante tres métodos (Tabla 3). Tras el normalizado, vemos que la muestra 6, correspondiente a ratones diabéticos 5W ("Diabetic 5W"), ya no aparece con outliers por ninguno de los tres métodos. Sin embargo, aparecen tres muestras, "Healthy 1W", "Diabetic 3W" y "Trained_diabetic 3W" con una cruz, es decir, algún outlier, en ambas muestras por el tercer método. Como solo hay outliers en uno de los tres métodos, se consideran como válidos los datos y se seguirá trabajando con ellos.

Tabla 3. Análisis de calidad mediante 3 métodos de los datos de microarray normalizados. En aquellos test donde se encontró un outlier se representó con una cruz.

array	sampleNames	*1	*2	*3	Group	HealthStatus	Trained	Time	ShortName
1	Healthy_1W			x	Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
2	Healthy_3W				Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
3	Healthy_5W				Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
4	Diabetic_1W				Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
5	Diabetic_3W			x	Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
6	Diabetic_5W				Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
7	Trained_diabetic_1W				Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
8	Trained_diabetic_3W			x	Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
9	Trained_diabetic_5W				Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W
10	Trained_1W				Healthy.Trained	Healthy	Yes	1_Week	Trained_1W
11	Trained_3W				Healthy.Trained	Healthy	Yes	3_Week	Trained_3W
12	Trained_5W				Healthy.Trained	Healthy	Yes	5_Week	Trained_5W

Si hacemos un diagrama de cajas y bigotes (Figura 7) vemos que los datos de cada una de las muestras tienen una distribución muy similar.

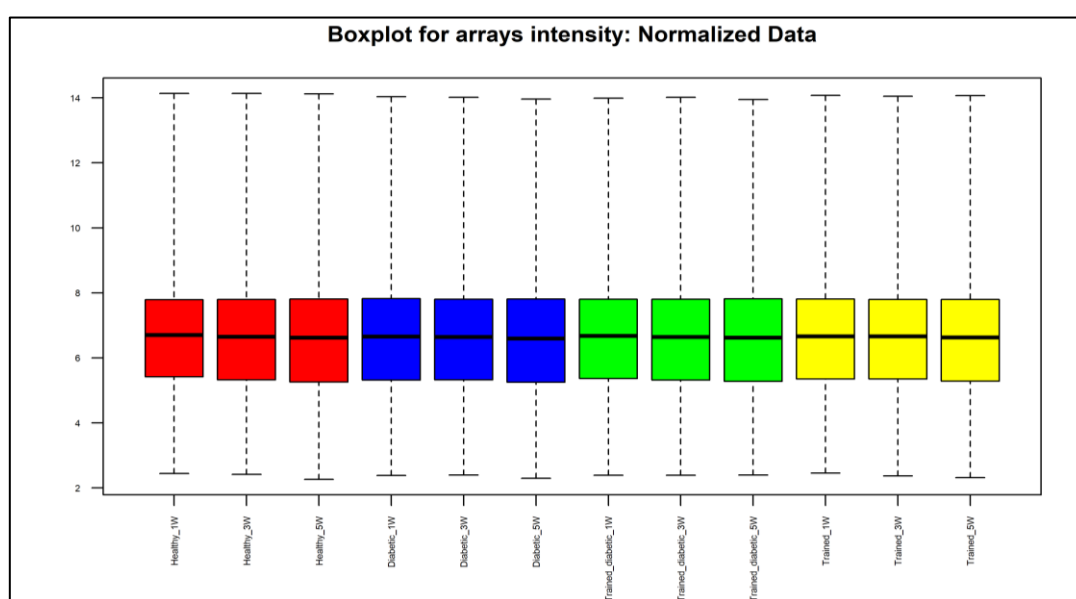


Figura 7. Diagrama de cajas y bigotes correspondientes a los 12 microarrays estudiados tras la normalización de los datos.

En este paso también hemos realizado un análisis de componentes principales, pero en esta ocasión usando los datos normalizados (Figura 8). En primera instancia, vemos que el primer componente, PC1, acumula el 50% de la varianza, un valor más alto que en el caso anterior. Por otro lado, PC2 acumula el 24%, algo menos que en el caso anterior. El primer componente, aunque de difícil interpretación, parece separar las muestras por tiempo en semanas, indistintamente de si el ratón es diabético, ha realizado deporte o ninguna de las dos cosas. Las muestras de microarray que proceden de RNA de ratones más viejos (5 semanas de duración del experimento), se sitúan en sentido positivo del eje X, mientras que las de ratones en los que el experimento duró 1 semana se sitúan en sentido negativo del eje. Siguiendo esta tendencia, quedan en una posición intermedia las muestras obtenidas de ratones que tuvieron 3 semanas de experimento. El hecho de encontrar diferencias en esta variable es un hecho que discutiremos más adelante. Pasando al PC2, vemos que las muestras se agrupan por estado de salud, quedando en el sentido positivo del eje los ratones control ("healthy"), independientemente de haber sido entrenados y de las semanas de entrenamiento. En sentido negativo, se agrupan por tanto las muestras procedentes de ratones con diabetes.

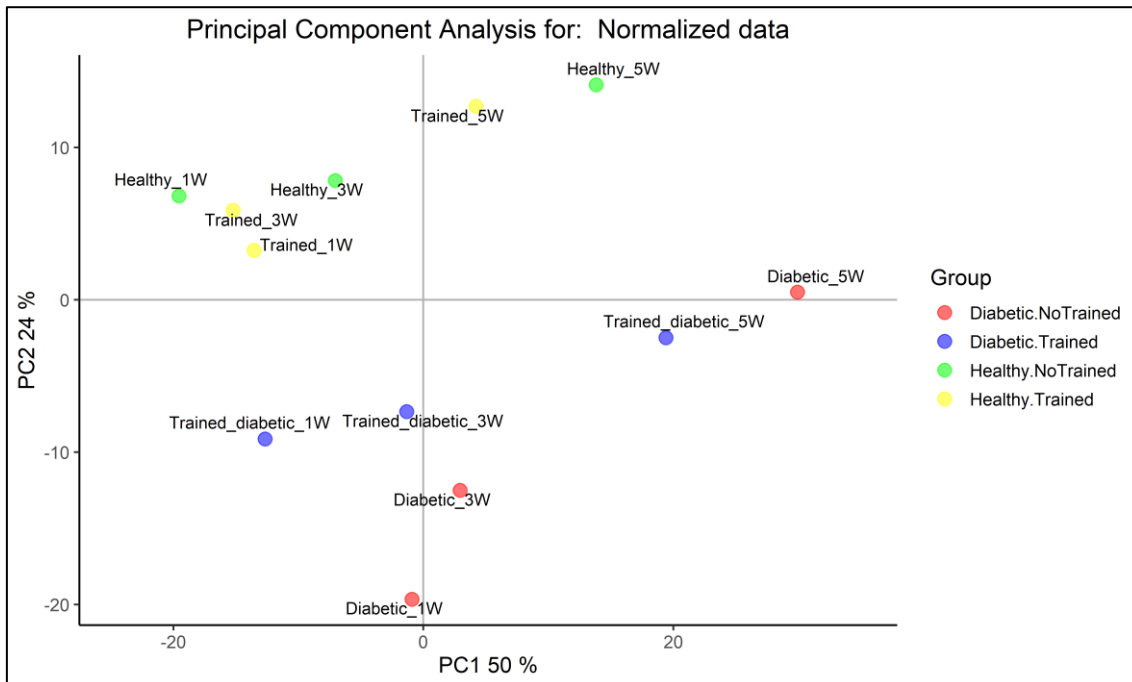


Figura 8. Análisis de componentes principales para las 12 muestras de microarray analizadas tras la normalización de los datos.

3.4.PASO 4: Detección de efectos derivados del "batch"

A continuación fue realizado un Principal Variance Component Analysis (PVCA) para buscar efectos producidos por el "batch", es decir, se espera detectar aquellas diferencias entre muestras producidas por causas que no sean las que estamos estudiando (ej. distintos lotes usados en el procesamiento del RNA, distinto técnico...). Hemos establecido un límite "threshold" de 0.6. Se representaron gráficamente a través de un diagrama de barras (Figura 9) las distintas fuentes de variación incluidas en nuestro análisis y el porcentaje de variabilidad atribuible. Se hace patente que la principal fuente de variación es el tiempo (1,3 o 5 semanas) en el que se ha tenido a los ratones en experimentación. La segunda fuente de más variación es el estado de salud.

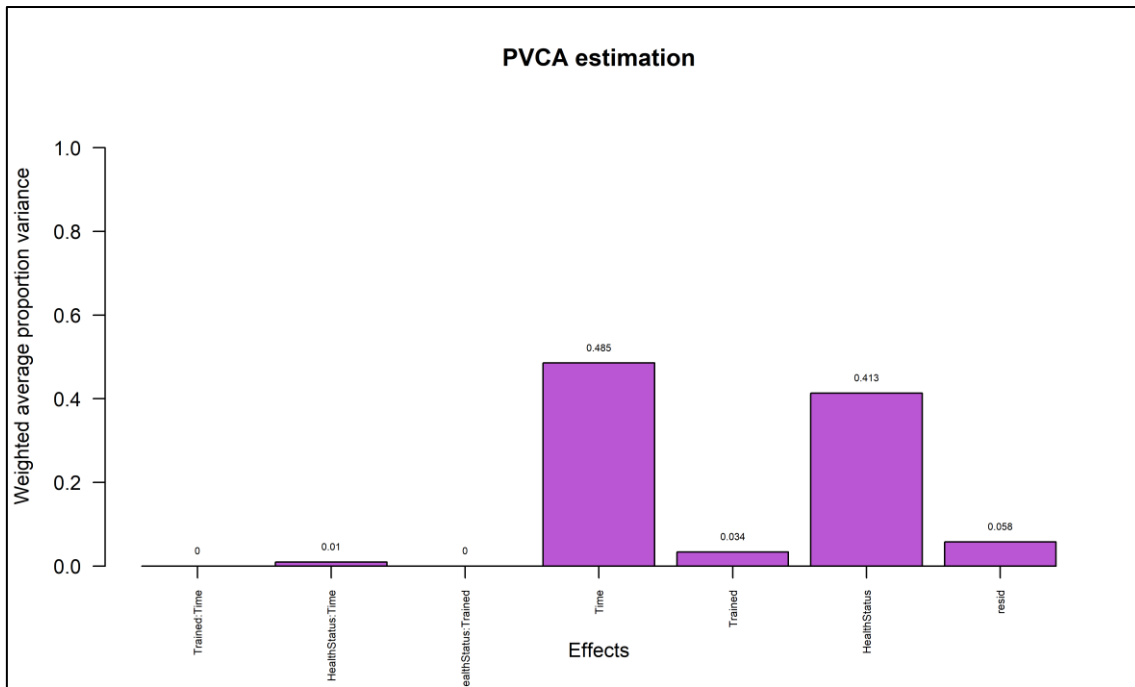


Figura 9. Diagrama de barras donde se observa el porcentaje de variabilidad atribuible a cada una de las variables estudiadas, solas o en combinación.

3.5. PASO 5: Detección de los genes más y menos variables y filtraje

Empezamos con los genes más variables entre muestras. En la siguiente gráfica (Figura 10) aparecen aquellos los genes más variables con una desviación estándar superior al 90-95%.

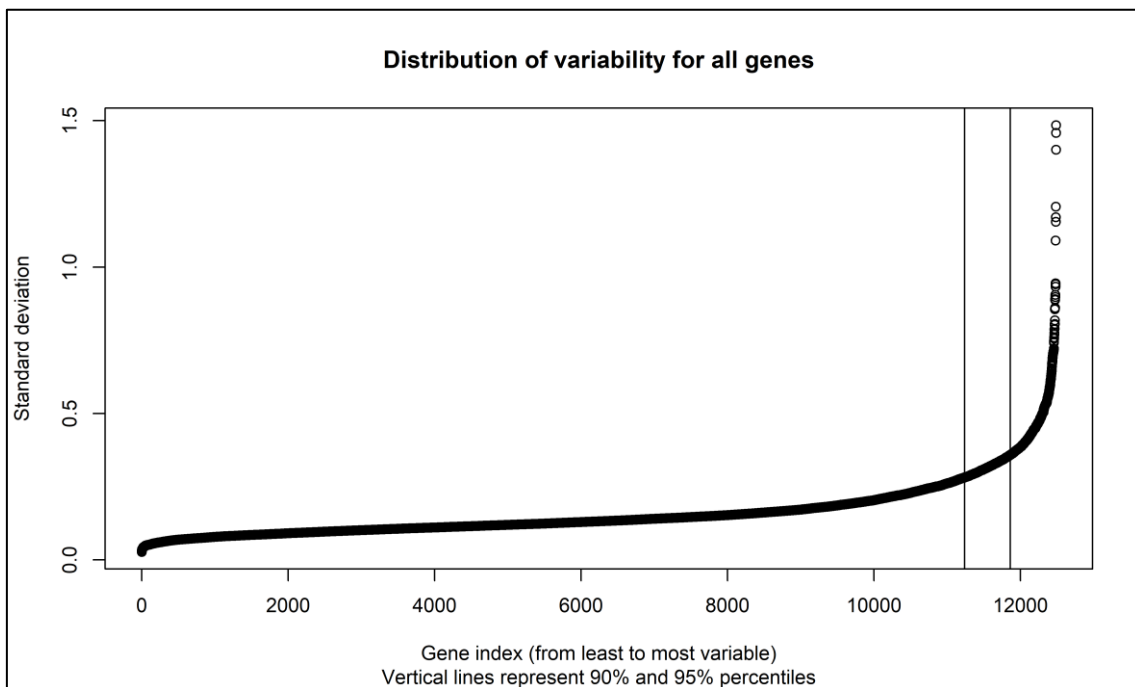


Figura 10. Distribución de la variabilidad de los genes. Los genes más variables se sitúan a la derecha.

Posteriormente, se realizó el filtraje de aquellos genes cuya variabilidad es posible atribuirla a la variación aleatoria más que a diferencias entre situaciones de nuestro experimento. El umbral de variabilidad definido fue de 0.75. La anotación de los datos de microarray que estamos manejando es la de "mgu74av2.db".

Tras eliminar y excluir los genes que no nos interesan (Figura 11) , nos hemos quedado con 2176 genes después del filtrado para analizar.

```
$numDupsRemoved
[1] 2639

$numLowVar
[1] 6526

$numRemoved.ENTREZID
[1] 1125

$feature.exclude
[1] 22
```

Figura 11. Número de genes eliminados en el filtraje.

3.6. PASO 6: Matriz de diseño y matriz de contraste

Lo primero que se realizó es la matriz de diseño. Está formada por 12 filas (1 por cada muestra) y 4 columnas ya que hemos considerado nuestro experimento como un experimento con un solo factor y cuatro niveles (Figura 12). Cada columna es un nivel.

	Diabetic.NoTrained	Diabetic.Trained	Healthy.NoTrained	Healthy.Trained
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0
7	0	1	0	0
8	0	1	0	0
9	0	1	0	0
10	0	0	0	1
11	0	0	0	1
12	0	0	0	1

```
attr("assign")
[1] 1 1 1 1
attr("contrasts")
attr("contrasts")$Group
[1] "contr.treatment"
```

Figura 12. Matriz de diseño.

A continuación se presenta la matriz de contrastes (Figura 13). Teniendo en cuenta los objetivos de este trabajo, los contrastes que se realizarán son los siguientes: (1) Ratón sano vs sano con entrenamiento, (2) Ratón diabético vs ratón sano, (3) Ratón diabético con entrenamiento vs ratón sano, (4) Ratón diabético entrenado vs ratón diabético y (5) La interacción entre estado de salud y entrenamiento ("INT").

Levels	Contrasts				
	TrainedvsNoTrained.Healthy	DiabeticvsControl.NoTrained	DiabeticTrainedvsControl	TrainedcvvsNoTrained.Diabetic	INT
Diabetic.NoTrained	0	1	0	-1	-1
Diabetic.Trained	0	0	1	1	1
Healthy.NoTrained	-1	-1	-1	0	1
Healthy.Trained	1	0	0	0	-1

Figura 13. Matriz de contrastes.

3.7. PASO 7: Selección de genes diferencialmente expresados

Se utilizaron modelos de Bayes empíricos para combinar la información de la matriz y los genes y mejorar así las estimaciones del error. Asimismo, se empleó el método de Benjamini-Hochberg para ajustar los p-valores obtenidos de forma que se pudiera controlar la tasa de falsos positivos.

A continuación, se obtuvo la lista de genes diferencialmente expresados en cada una de las comparaciones ordenados de más a menos diferencialmente expresados. Se mostrarán los 5 genes más diferencialmente expresados con sus correspondientes valores de logFoldChange (logFC), expresión media (AveExpr), valor t, p-valor, p-valor ajustado, valor B (Tablas 4,5,6,7,8).

Tabla 4. Top 5 genes diferencialmente expresados en la comparación 1: Ratón sano vs sano con entrenamiento.

	logFC <dbl>	AveExpr <dbl>	t <dbl>	P.Value <dbl>	adj.P.Val <dbl>	B <dbl>
100921_at	-1.6954610	6.947136	-7.828799	3.413894e-06	0.007428633	-0.2858227
101071_at	-1.8109932	5.483442	-5.635183	9.083567e-05	0.098829213	-1.1743220
98569_at	-0.5198317	9.296581	-2.932546	1.197232e-02	0.999521589	-3.1368634
100593_at	-0.8750638	5.957940	-2.891331	1.294597e-02	0.999521589	-3.1741444
160901_at	-0.7890120	6.087163	-2.805927	1.521973e-02	0.999521589	-3.2517406

Tabla 5. Top 5 genes diferencialmente expresados en la comparación 2: Ratón diabético vs sano.

	logFC <dbl>	AveExpr <dbl>	t <dbl>	P.Value <dbl>	adj.P.Val <dbl>	B <dbl>
95731_at	2.322352	8.008379	12.456348	1.845874e-08	4.016622e-05	9.388157
160522_at	-1.536395	8.954212	-9.457259	4.351244e-07	4.734154e-04	6.700336
94297_at	1.984398	9.089250	8.345269	1.721765e-06	1.248854e-03	5.456739
100921_at	-1.740535	6.947136	-8.036927	2.581357e-06	1.404258e-03	5.083766
102967_at	-1.929720	6.803440	-7.185601	8.370910e-06	3.491158e-03	3.985178

Tabla 6. Top 5 genes diferencialmente expresados en la comparación 3: Ratón diabético con entrenamiento vs ratón sano.

	logFC <dbl>	AveExpr <dbl>	t <dbl>	P.Value <dbl>	adj.P.Val <dbl>	B <dbl>
95731_at	1.673315	8.008379	8.975122	7.776875e-07	0.001692248	6.014329
100921_at	-1.640036	6.947136	-7.572874	4.848379e-06	0.003233014	4.413217
94297_at	1.768267	9.089250	7.436343	5.865241e-06	0.003233014	4.242516
160522_at	-1.206558	8.954212	-7.426954	5.943041e-06	0.003233014	4.230676
96221_at	1.049813	7.175732	6.748565	1.586386e-05	0.006903950	3.339623

Tabla 7. Top 5 genes diferencialmente expresados en la comparación 4: Ratón diabético entrenado vs ratón diabético.

	logFC <dbl>	AveExpr <dbl>	t <dbl>	P.Value <dbl>	adj.P.Val <dbl>	B <dbl>
162015_f_at	0.8177101	9.547900	3.599808	0.003378752	0.9988969	-3.918268
95731_at	-0.6490371	8.008379	-3.481226	0.004226301	0.9988969	-3.946285
103084_at	-0.9001082	10.238450	-3.442440	0.004548028	0.9988969	-3.955625
98089_at	-0.5529877	7.725916	-3.345733	0.005462474	0.9988969	-3.979287
94534_at	0.4813928	9.563225	3.289206	0.006080818	0.9988969	-3.993362

Tabla 8. Top 5 genes diferencialmente expresados en la comparación 5: Interacción.

	logFC <dbl>	AveExpr <dbl>	t <dbl>	P.Value <dbl>	adj.P.Val <dbl>	B <dbl>
100921_at	1.7959597	6.947136	5.863932	6.258452e-05	0.1361839	-3.031140
101071_at	1.5922214	5.483442	3.503319	4.053454e-03	0.9988892	-3.652202
98569_at	0.6760587	9.296581	2.696817	1.870404e-02	0.9988892	-3.962202
103084_at	-0.9226285	10.238450	-2.495075	2.731332e-02	0.9988892	-4.045538
94817_at	0.5941550	7.538252	2.421032	3.135062e-02	0.9988892	-4.076451

Como podemos observar, la primera columna de las tablas obtenidas en las cinco comparaciones corresponde al ID de cada conjunto de sondas de Affymetrix. Para poder conocer los genes diferencialmente expresados necesitamos conocer qué gen corresponde a cada ID (anotación). Se ha utilizado la anotación “mgu74av2.db” para anotar los genes y se han generado 5 archivos .csv correspondientes a las 5 comparaciones con la información contenida como en la Tabla 9.

Tabla 9. Tabla de asociación para los genes y sus valores de expresión en la comparación 1.

PROBEID	SYMBOL	ENTREZID	GENENAME	logFC	AveExpr	t	P.Value	adj.P.Val
1 100011_at	Klf3	16599	Kruppel-like factor 3 (basic)	0.02931764	5.044166	0.1645098	0.8719351	0.9995216
2 100017_at	Mybph	53311	myosin binding protein H	0.08898649	7.765299	0.3336887	0.7440930	0.9995216
3 100022_at	Cish	12700	cytokine inducible SH2-containing protein	-0.24595596	8.434054	-1.5360801	0.1491992	0.9995216
4 100032_at	Sp1	20683	trans-acting transcription factor 1	0.21557721	6.853628	0.5961384	0.5616217	0.9995216
5 100037_at	Ddx18	66942	DEAD (Asp-Glu-Ala-Asp) box polypeptide 18	0.07154840	5.823712	0.4226695	0.6796480	0.9995216
6 100041_at	Slc25a39	68066	solute carrier family 25, member 39	0.17715972	8.540386	1.3362499	0.2050588	0.9995216

3.7.1. Volcano plots

Una vez tenemos las anotaciones hechas, hemos representado mediante "volcano plots" los genes diferencialmente expresados en los distintos grupos, resaltando los 5 más diferencialmente expresados.

Si nos fijamos en el primer volcano plot (Figura 14), que corresponde a la comparación ratón sano vs sano con entrenamiento, vemos que solo existen dos genes por debajo de -1 Log2 Fold Change, *Tnni3* y *Myh6*. No hay apenas genes muy diferencialmente expresados, con un valor alto de Log2 Fold Change entre ratones sanos que han realizado entrenamiento de aquellos que no.

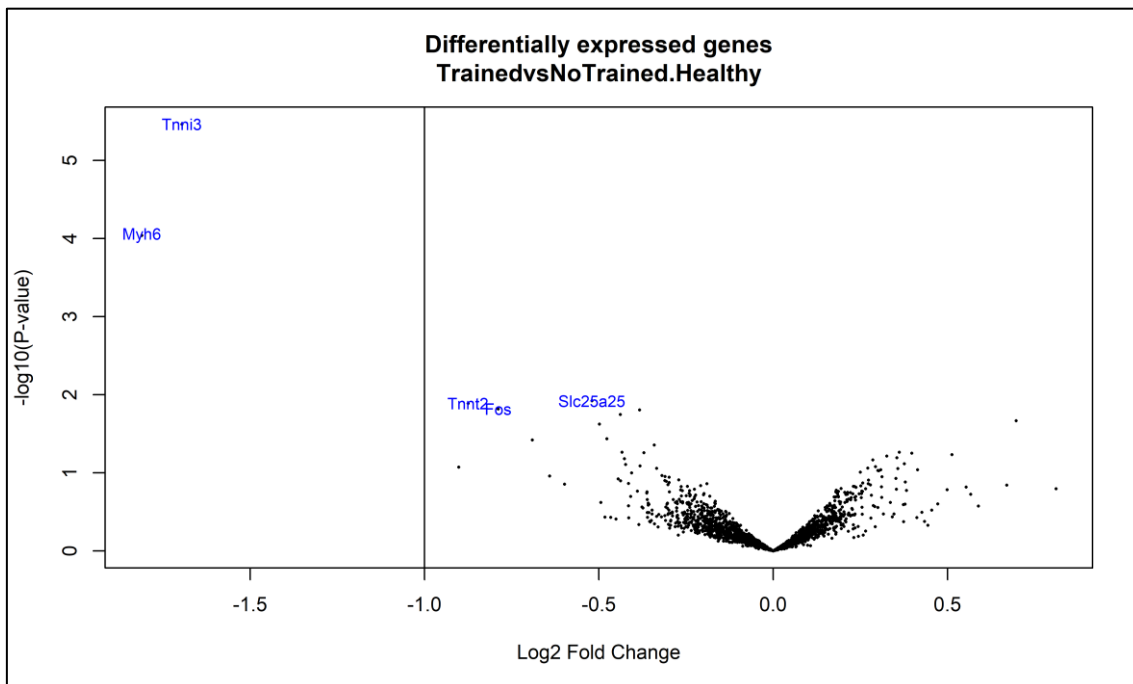


Figura 14. Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 1: Ratón sano vs sano con entrenamiento.

Cuando miramos el volcano plot con los top 5 genes más diferencialmente expresados en la comparación 2 (Figura 15), vemos que en esta ocasión existen más genes con un valor alto de Log2 Fold Change entre ratones diabéticos y control que entre control con y sin ejercicio. Estos genes más diferencialmente expresados son: *Tnni3*, *Nrep*, *Gdap1*, *Sesn1* y *Fkpb5*.

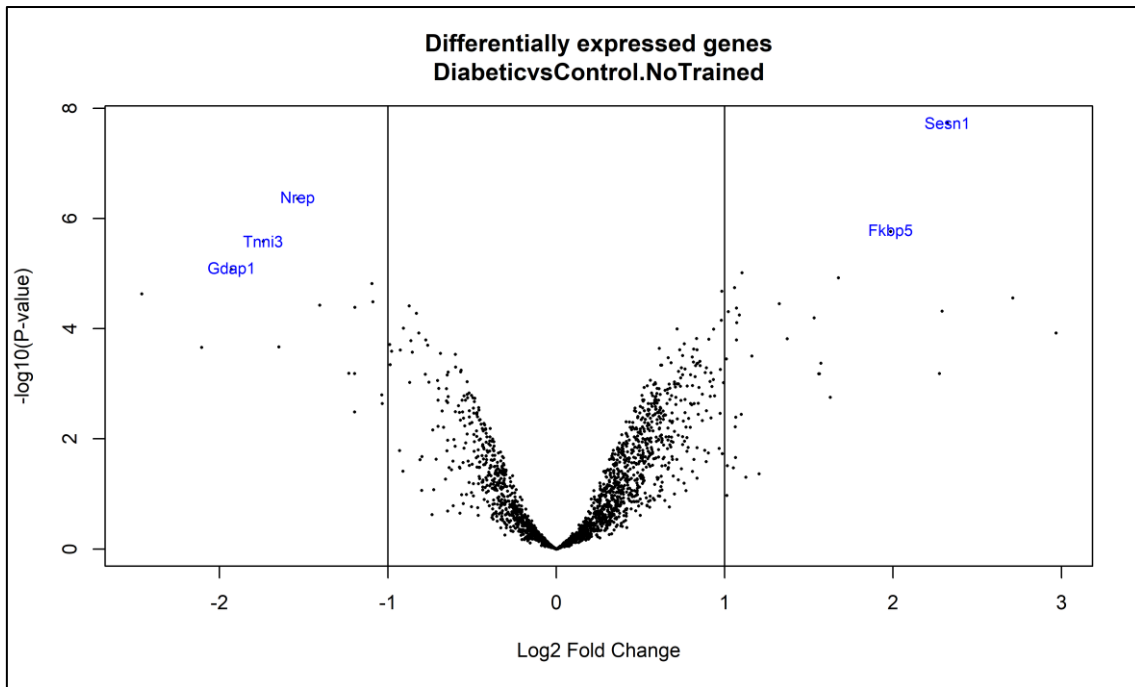


Figura 15. Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 2: Ratón diabético vs sano.

En la Figura 16 tenemos el volcano plot con los top 5 genes más diferencialmente expresados en la comparación 3: Ratón diabético con entrenamiento vs ratón sano. De nuevo vemos que existen más genes con un valor alto de Log2 Fold Change entre ratones diabéticos entrenados y control que en la comparación primera. Resulta interesante que de los top 5 genes, 4 (*Tnni3*, *Nrep*, *Sesn1* y *Fkbp5*) son los mismos que aparecen más diferencialmente expresados entre ratones diabéticos sin ejercicio vs ratón sano. El otro es *Tra3fp2*.

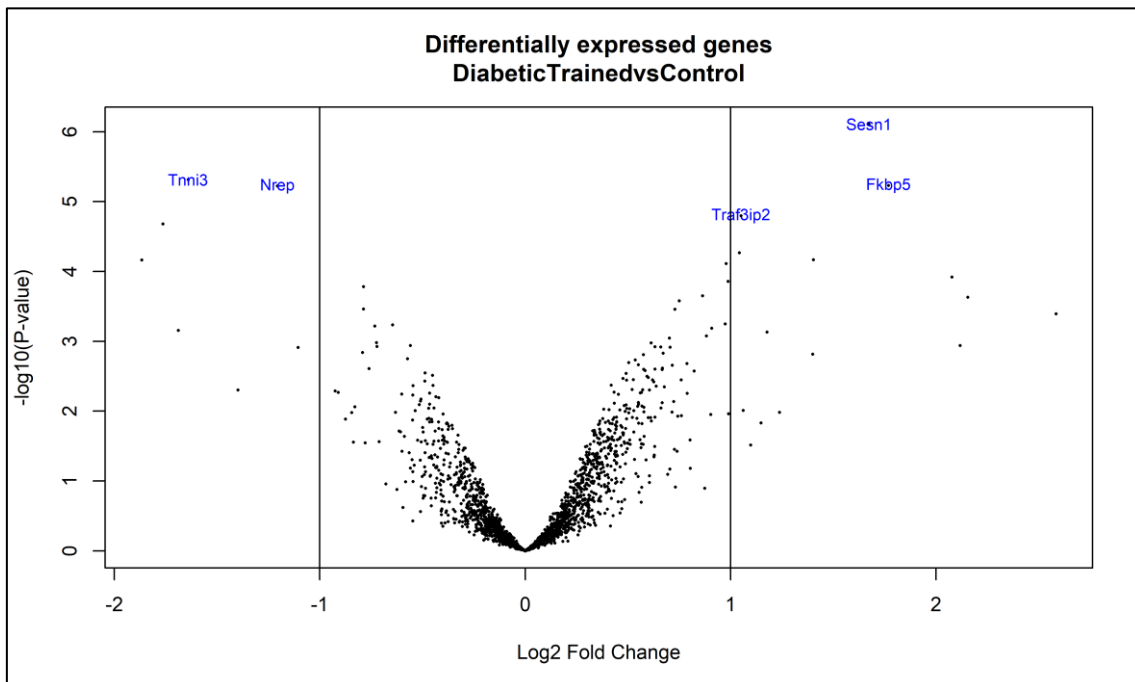


Figura 16. Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 3: Ratón diabético con entrenamiento vs sano.

Por otro lado, cuando nos fijamos en los genes más diferencialmente expresados en la comparación 4: Ratón diabético con entrenamiento vs ratón diabético sin entrenamiento, vemos que no existe ningún gen con un valor de log2 Fold Change por encima de 1 o por debajo de -1 (Figura 17).

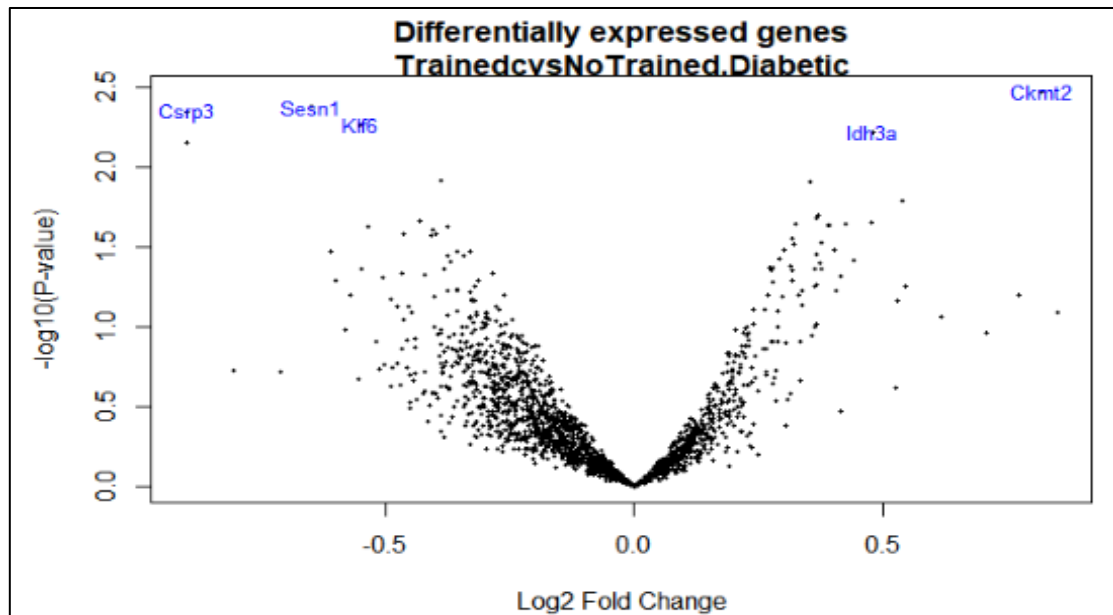


Figura 17. Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 4: Ratón diabético entrenado vs diabético no entrenado.

Finalmente, en la Figura 18, tenemos el volcano plot con los genes más diferencialmente expresados en la interacción.

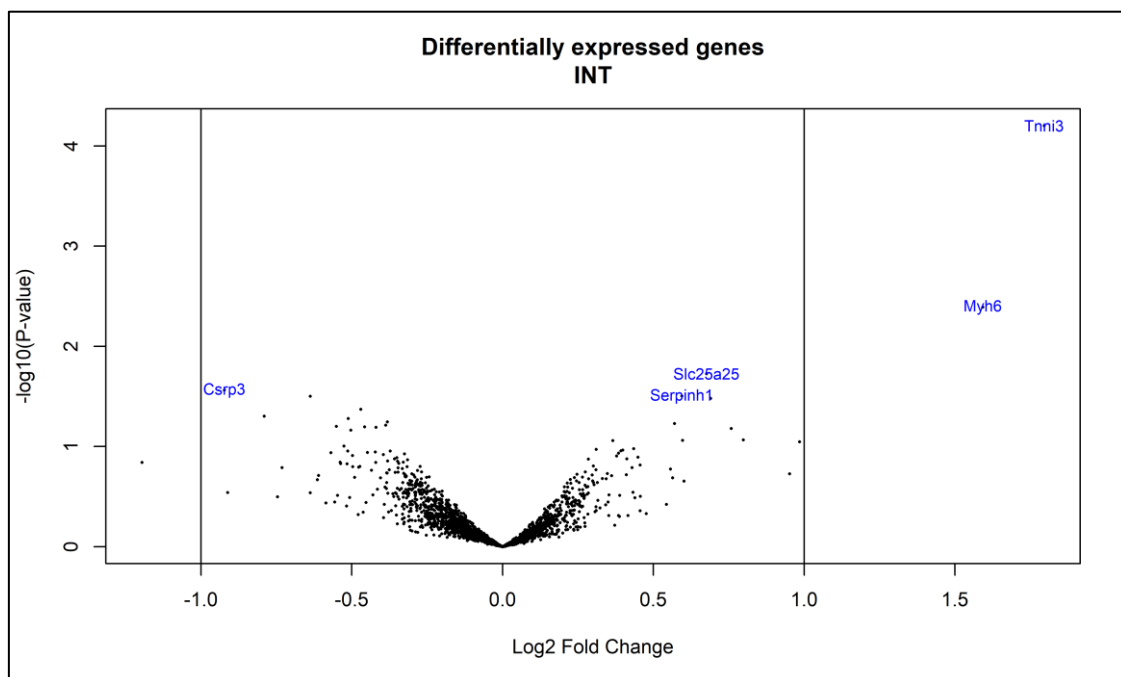


Figura 18. Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 5: Interacción.

3.7.2. Comparaciones múltiples y diagrama de Venn

El siguiente paso fue estudiar qué genes están up-regulados y down-regulados en cada una de las comparaciones. El número de genes en cada situación se encuentran en la Tabla 10.

Tabla 10. Genes up-regulados y down-regulados en las distintas comparaciones.

	TrainedvsNoTrained.Healthy	DiabeticvsControl.NoTrained	DiabeticTrainedvsControl	TrainedcvsNoTrained.Diabetic	INT
Down	2	15	6	0	0
NotSig	2174	2135	2159	2176	2176
Up	0	26	11	0	0

A continuación, se representó mediante un diagrama de Venn los genes diferencialmente expresados en común entre las 4 categorías estudiadas en este trabajo. Se han considerado aquellos genes con un valor de $FDR < 0.1$ y $\log FC > 1$. Como vemos, no existe ningún gen compartido entre las 4 categorías (Figura 19).

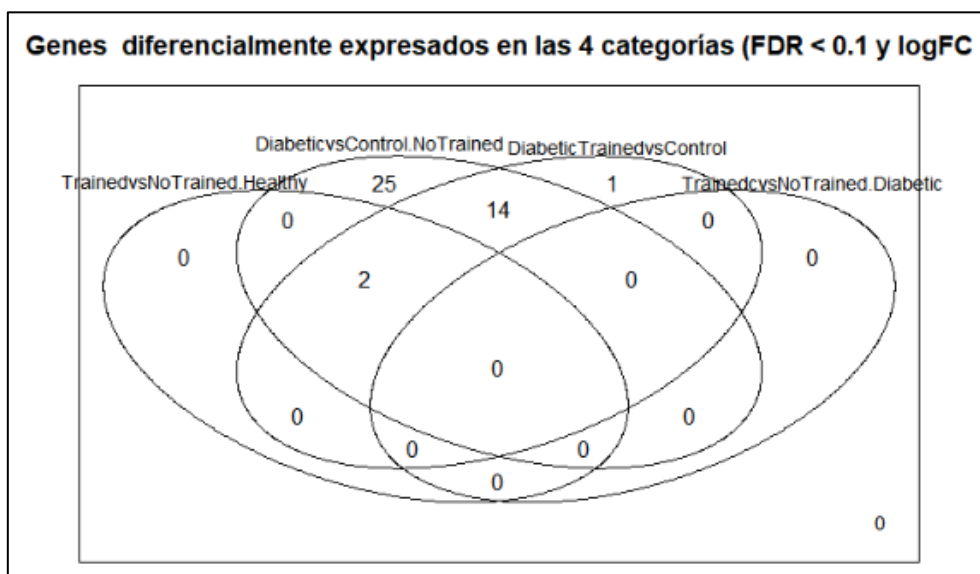


Figura 19. Diagrama de Venn con los genes diferencialmente expresados en común en las distintas comparaciones de ratones estudiadas.

3.7.3. Visualización de los perfiles de expresión usando mapas de calor

Lo siguiente que se hizo para estudiar los genes diferencialmente expresados fue realizar dos mapas de calor (también llamados "heatmaps"), uno sin agrupación de muestras y otro con las muestras y genes agrupados por clusters. A través de estos heatmaps podemos ver los genes que están up-regulados y down-regulados.

En la Figura 20 tendríamos el heatmap sin ningún tipo de agrupación de muestras. A la derecha podemos ver el listado de los genes que se encuentran up/downregulados.

Por otro lado, en la Figura 21 tendríamos el heatmap con dos tipos de agrupamientos: arriba estaría la agrupación entre muestras (columnas) y a la izquierda la agrupación por genes (filas). Lo primero que se puede observar es que las muestras de microarray se separan claramente por el estado de salud del ratón, quedando por un lado agrupadas todas las muestras de ratones sanos y por otra las de ratones diabéticos, independientemente de realizar o no ejercicio.

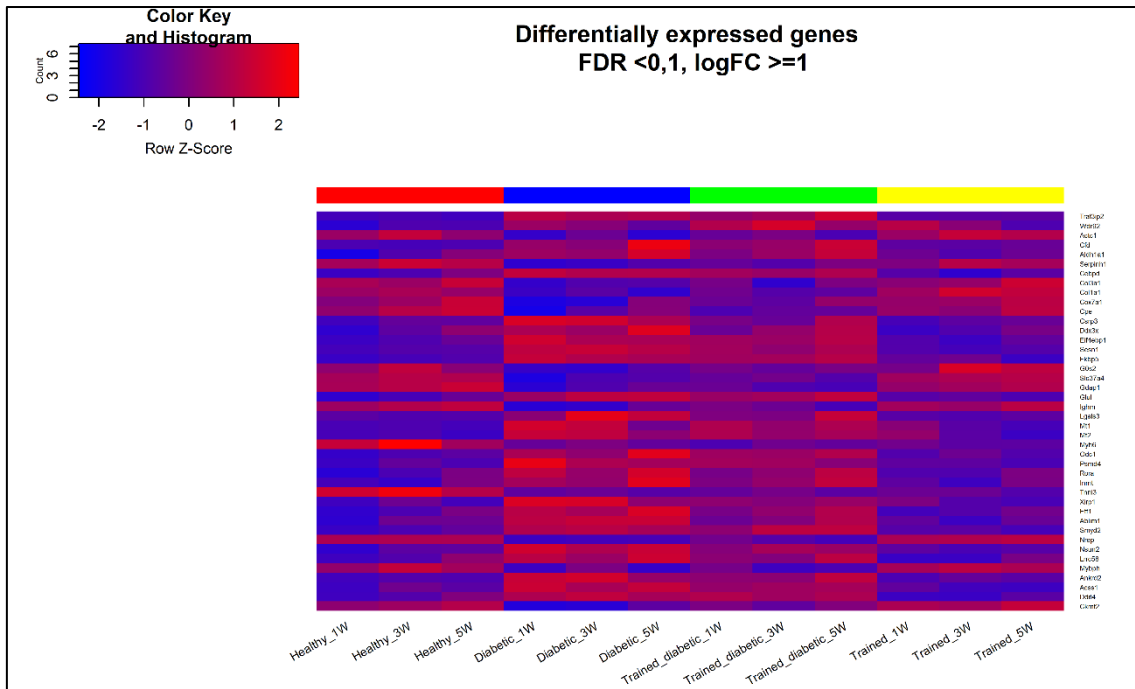


Figura 20. Heatmap con los genes diferencialmente expresados en las distintas muestras.

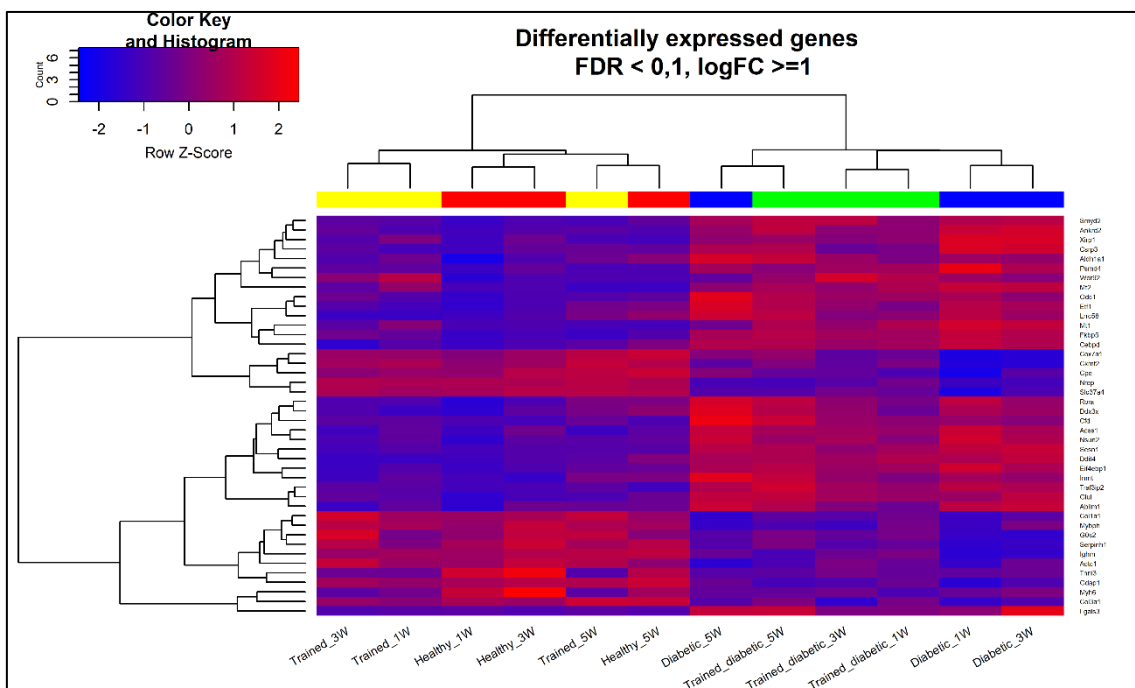


Figura 21. Heatmap con los genes diferencialmente expresados en las distintas muestras.

Si nos fijamos en el comportamiento de los genes entre ratones sanos y diabéticos podríamos agruparlos en 6 clusters:

- Cluster 1º: *Symd2*, *Ankrd2*, *Xirp1*, *Csrp3*, *Aldh1a1*, *Pmsd4*, *Wdr92*, *Mt2*, *Odc1*, *Etf1*, *Lrrc58*, *Mt1*, *Fkbp5*, *Cebpd* → Genes up-regulados (se han transcrito más) en ratones diabéticos que en ratones sanos. Donde menos están expresados es en ratones sanos sin ejercicio y experimento duración 1 semana.

- Cluster 2º: *Cox7a1, Ckmt2, Cpe, Nrep, Slc37a4* → Genes up-regulados en ratones sanos que en diabéticos. Tienen su máximo de expresión en los ratones diabéticos tanto sin y con entrenamiento, pero ambos con duración del experimento 5 semanas.
- Cluster 3º: *Rora, Ddx3x, Cfd, Acss1, Nsun2, Sesn1, Ddit4, Eif4ebp1, Inmt, Traf3ip2, Glul, Ablim1* → Genes up-regulados (se han transcrito más) en ratones diabéticos que en ratones sanos. Donde menos están expresados es en ratones sanos sin ejercicio y experimento duración 1 semana y donde más en los ratones diabéticos sin entrenamiento con duración del experimento 5 semanas.
- Cluster 4º: *Col1a1, Mybph, G0s2, Serpini, Ighm, Actc1* → Genes downregulados en ratones diabéticos.
- Cluster 5º: *Tnni3, Gdap1, Myh6, Col3a1* → Genes upregulados en ratones sanos. *Tnni3* y *Myh6* están muy expresados en ratones sanos que no han sido sometidos a entrenamiento y muy downregulados (igual que en diabéticos) en ratones que han sido sometidos a entrenamiento.
- Cluster 6º: *Lgals3* → Gen upregulado en ratones diabéticos, sobre todo en aquellos con 5 semanas de experimento.

3.8. PASO 8: Análisis de significación biológica

El siguiente paso de este trabajo fue realizar un análisis de significación biológica para ver si entre la lista de genes diferencialmente expresados, había procesos biológicos, funciones moleculares o rutas metabólicas que se encontraban enriquecidas entre una condición y otra. El punto de corte para incluir a los genes fue de $FDR < 0,15$.

Para ello, se realizó el análisis de enriquecimiento usándose las siguientes bases de datos de anotaciones: para procesos biológicos/funciones moleculares Gene Ontology (GO) y para rutas metabólicas Kyoto Encyclopedia of Genes and Genomes (KEGG).

En la Figura 22 se recoge el nº de genes que fueron analizados en cada comparación. Dado que en la comparación 4, ratones diabéticos con y sin entrenamiento, no hay genes que hayan pasado nuestro punto de corte de expresión diferencial no se comentará esta categoría. Por ende, las diferencias a nivel genético entre ratones diabéticos, independientemente de que hayan sido sometidos a un programa de entrenamiento o no.

TrainedvsNoTrained.Healthy	DiabeticvsControl.NoTrained
2	597
DiabeticTrainedvsControl	TrainedcvvsNoTrained.Diabetic
81	0
INT	
1	

Figura 22. Nº de genes seleccionados para el estudio de significación biológica.

El análisis de significación biológica solo se hizo para las 3 primeras comparaciones. Para dicho estudio vamos a considerar como nuestro “universe” todos los genes de los anteriores que tienen al menos una anotación en Gene Ontology y un punto de corte para el p-valor de 0.05.

Como podemos observar en la Figura 23, el análisis de significación biológica realizado en la comparación 1, mostró que existe una disminución de la transcripción en genes implicados en la contracción muscular, la homeostasis iónica y la conducción cardíaca cuando el ratón sano está entrenado vs sin entrenamiento.

## Comparison: TrainedvsNoTrained.Healthy						
##	ID		Description	GeneRatio	BgRatio	
##	R-MMU-390522	R-MMU-390522	Striated Muscle Contraction	2/2	33/8772	
##	R-MMU-397014	R-MMU-397014	Muscle contraction	2/2	177/8772	
##	R-MMU-5578775	R-MMU-5578775	Ion homeostasis	1/2	52/8772	
##	R-MMU-5576891	R-MMU-5576891	Cardiac conduction	1/2	123/8772	
##		pvalue	p.adjust	qvalue	geneID	Count
##	R-MMU-390522	1.372512e-05	5.490048e-05	NA	Tnni3/Myh6	2
##	R-MMU-397014	4.048911e-04	8.097821e-04	NA	Tnni3/Myh6	2
##	R-MMU-5578775	1.182144e-02	1.576192e-02	NA	Tnni3	1
##	R-MMU-5576891	2.784874e-02	2.784874e-02	NA	Tnni3	1

Figura 23. Procesos biológicos enriquecidos en la comparación 1: Ratón sano vs sano con entrenamiento.

Asimismo, cuando se compara el ratón diabético con el control, ambos sin haber sido entrenados, se observa un enriquecimiento de genes implicados en funciones relacionadas con la síntesis de proteínas (Figura 24), como *Rps3*, *Rps5*, *Rps7*, *Rps10*... en el ratón sano.

## Comparison: DiabeticvsControl.NoTrained						
##	ID					
##	R-MMU-72613	R-MMU-72613				
##	R-MMU-72737	R-MMU-72737				
##	R-MMU-72706	R-MMU-72706				
##	R-MMU-72689	R-MMU-72689				
##	R-MMU-156827	R-MMU-156827				
##	R-MMU-72766	R-MMU-72766				
##	Description					
##	R-MMU-72613	Eukaryotic Translation Initiation				
##	R-MMU-72737	Cap-dependent Translation Initiation				
##	R-MMU-72706	GTP hydrolysis and joining of the 60S ribosomal subunit				
##	R-MMU-72689	Formation of a pool of free 40S subunits				
##	R-MMU-156827	L13a-mediated translational silencing of Ceruloplasmin expression				
##	R-MMU-72766	Translation				
##	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	
##	R-MMU-72613	28/356	112/8772	3.316585e-15	1.190654e-12	9.303894e-13
##	R-MMU-72737	28/356	112/8772	3.316585e-15	1.190654e-12	9.303894e-13
##	R-MMU-72706	24/356	105/8772	2.753523e-12	6.590099e-10	5.149572e-10
##	R-MMU-72689	22/356	94/8772	1.385929e-11	2.367913e-09	1.850312e-09
##	R-MMU-156827	23/356	104/8772	1.648965e-11	2.367913e-09	1.850312e-09
##	R-MMU-72766	33/356	217/8772	3.533560e-11	4.228494e-09	3.304189e-09
##	geneID					
##	R-MMU-72613	Eif4ebp1/Rpl18/Rps23/Rps4x/Eif4b/Rpl6/8/Eif2b5/Eif3g/Eif5b/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1/				
##	R-MMU-72737	Eif4ebp1/Rpl18/Rps23/Rps4x/Eif4b/Rpl6/8/Eif2b5/Eif3g/Eif5b/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1/				
##	R-MMU-72706	Rpl18/Rplp0/Rps18/Eif3g/Eif5b/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/R				
##	R-MMU-72689	s19/Eif3c/Rplp0/Rps18/Eif3g/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/R				
##	R-MMU-156827	R				
##	R-MMU-72766	Eif4ebp1/Rpl18/Etf1/Eef2/Rps23/Rps4x/Eif4b/Rpl6/Rpl18/Eef1a1/Eif2b1/5/Eif3g/Eif5b/Eif3l/Rps5/Mrpl45/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1/				
##	Count					
##	R-MMU-72613	28				
##	R-MMU-72737	28				
##	R-MMU-72706	24				
##	R-MMU-72689	22				
##	R-MMU-156827	23				

Figura 23. Procesos biológicos enriquecidos en la comparación 2: Ratón diabético vs sano.

Por otro lado, cuando comparamos las muestras procedentes de ratones diabéticos pero que han sido entrenados con ratones control sin entrenamiento, vemos que existe una disminución de la transcripción de genes implicados en la formación de las fibras de colágeno (Figura 24).

```
## #####
## Comparison: DiabeticTrainedvsControl
## ID
## R-MMU-1474290 R-MMU-1474290
## R-MMU-2022090 R-MMU-2022090
## R-MMU-1650814 R-MMU-1650814
## R-MMU-8948216 R-MMU-8948216
## R-MMU-2243919 R-MMU-2243919
## R-MMU-1442490 R-MMU-1442490
##
## Description
## R-MMU-1474290 Collagen formation
## R-MMU-2022090 Assembly of collagen fibrils and other multimeric structures
## R-MMU-1650814 Collagen biosynthesis and modifying enzymes
## R-MMU-8948216 Collagen chain trimerization
## R-MMU-2243919 Crosslinking of collagen fibrils
## R-MMU-1442490 Collagen degradation
##
## GeneRatio BgRatio pvalue p.adjust qvalue
## R-MMU-1474290 6/53 81/8772 8.349036e-06 0.002529758 0.002249846
## R-MMU-2022090 5/53 57/8772 2.191559e-05 0.002622532 0.002332354
## R-MMU-1650814 5/53 59/8772 2.596566e-05 0.002622532 0.002332354
## R-MMU-8948216 4/53 39/8772 8.353404e-05 0.006327703 0.005627556
## R-MMU-2243919 3/53 18/8772 1.594089e-04 0.009660180 0.008591301
## R-MMU-1442490 4/53 55/8772 3.225077e-04 0.016286640 0.014484558
##
## geneID Count
## R-MMU-1474290 Col15a1/Serpinh1/Coll1a1/Lox/Col4a1/Col3a1 6
## R-MMU-2022090 Col15a1/Coll1a1/Lox/Col4a1/Col3a1 5
## R-MMU-1650814 Col15a1/Serpinh1/Coll1a1/Col4a1/Col3a1 5
## R-MMU-8948216 Col15a1/Coll1a1/Col4a1/Col3a1 4
## R-MMU-2243919 Coll1a1/Lox/Col4a1 3
## R-MMU-1442490 Col15a1/Coll1a1/Col4a1/Col3a1 4
```

Figura 24. Procesos biológicos enriquecidos en la comparación 3: Ratón diabético con entrenamiento vs sano sin entrenamiento.

Con el código ejecutado en R se han generado dos tipos de archivo .pdf para cada comparación, uno llamado “ReactomePABarplot” y otro “ReactomePABcnetplot”. En el primero tenemos un diagrama de barras con las funciones biológicas (GO Terms) representadas según el p.adjusted value (un ejemplo podemos verlo en la Figura 25). En el segundo archivo tenemos la red de interacción entre los genes diferencialmente expresados que cumplían los puntos de corte definidos (un ejemplo podemos verlo en la Figura 26).

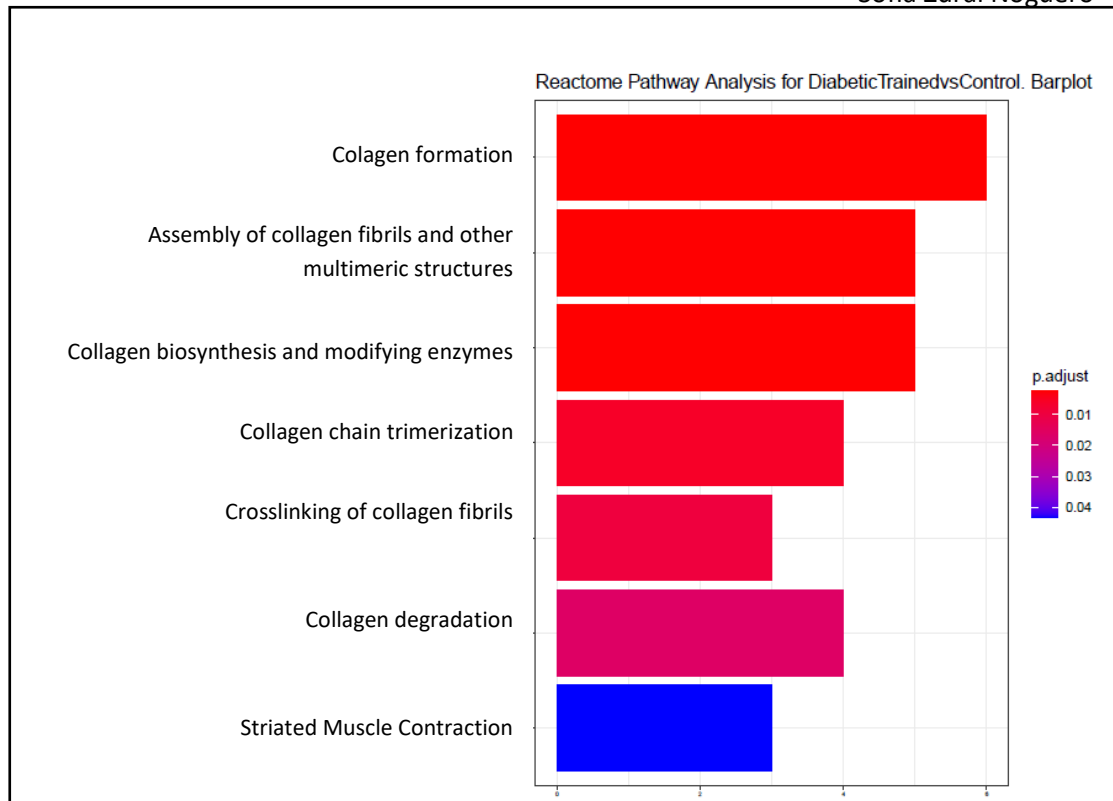


Figura 25. Diagrama de barras disponible en el fichero ReactomePABarplot donde se representan los procesos biológicos enriquecidos en la comparación 3 (Ratón diabético con entrenamiento vs sano sin entrenamiento) con sus respectivos p-adjusted values.

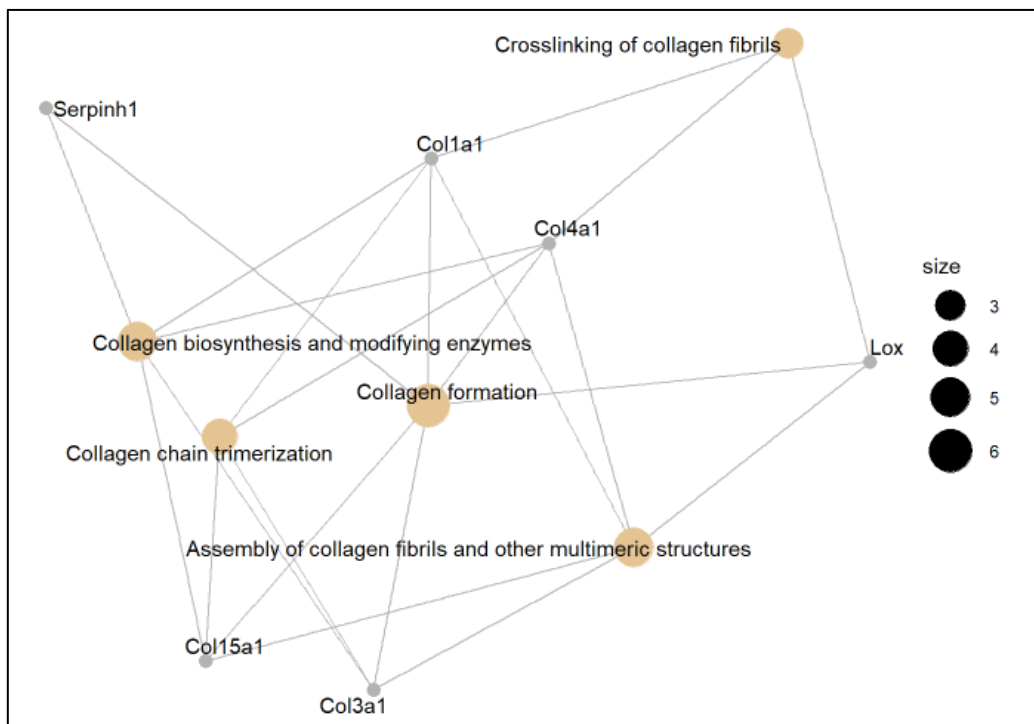


Figura 26. Red de interacción génica disponible en el fichero ReactomePABcnetplot donde se representan los procesos biológicos enriquecidos y los genes implicados en la comparación 3 (Ratón diabético con entrenamiento vs sano sin entrenamiento).

El archivo targets.csv, los ficheros .CEL, los archivos generados así como el código de R se encuentran en el repositorio de Github indicado en la primera página de este documento.

Discusión

En el presente trabajo se ha estudiado si la expresión génica era distinta en ratones según su estado de salud y tras haber sido sometidos a un programa de entrenamiento de duración 1,3 o 5 semanas. A través del RNA obtenido del músculo esquelético en los distintos grupos de ratones, se ha analizado con microarray y posteriormente estudiado con herramientas bioinformáticas.

Tras el normalizado de datos, lo primero que se observó tras un ACP, que los ratones se agrupaban por un lado según el tiempo de duración del experimento, y por otro lado se agrupaban según el estado de salud (sanos/diabéticos), independientemente de haber realizado ejercicio o no.

Cuando se evaluó el efecto del entrenamiento en ratones sanos, se observó una disminución de la transcripción en genes implicados en la contracción muscular, la homeostasis iónica y la conducción cardíaca. Los genes más diferencialmente expresados fueron *Tnni3* y *Myh6*. *Tnni3* sintetiza una de las tres subunidades del complejo de troponina en el músculo estriado, y mutaciones de pérdida de función en este gen están relacionadas con distintas cardiopatías (Huang et al. 2004). Asimismo, *Myh6* es un gen que está implicado en la síntesis de la subunidad alfa de la cadena pesada de la miosina y también ligado a cardiopatías y defectos en el tabique auricular (Ching et al. 2005).

Por otro lado, cuando se compararon las muestras procedentes de ratones diabéticos con las de ratones sanos se obtuvo un mayor número de genes diferencialmente expresados, entre los que destacan: *Tnni3*, *Nrep*, *Gdap1*, *Sesn1* y *Fkpb5*. Los tres primeros genes están up-regulados en ratones sanos, mientras que los dos últimos están más expresados en ratones diabéticos. *Tnni3* ya ha sido explicada su función. *Nrep* y *Gdap1* son genes implicados en el correcto desarrollo neuronal. Mutaciones en este último han sido relacionadas con el síndrome de Charcot-Marie-Tooth (Baxter et al. 2002). Asimismo, por mencionar un gen up-regulado en diabéticos, *Sesn1* forma parte de la familia de las sestrinas y su síntesis es inducida por *p53* en respuesta al daño celular y al estrés oxidativo (Budanov et al. 2008).

Resulta interesante resaltar que el análisis de significación biológica reveló que entre esos dos grupos existía una reducción de genes implicados en la síntesis proteica cuando se compararon ratones diabéticos *versus* sanos.

Cuando se evalúa el efecto del ejercicio en ratones diabéticos, al compararlos con ratones diabéticos sin ejercicio no se encuentran genes diferencialmente expresados a un nivel por encima del punto de corte establecido. Sin embargo, cuando se compara el ratón diabético que ha realizado ejercicio con el ratón sano sí encontramos varios genes diferencialmente expresados por encima de dicho punto de corte. Entre ellos, los anteriormente señalados, *Tnni3*, *Nrep* (up regulados en sanos), *Sesn1* y *Fkpb5* (up regulados en diabéticos entrenados). No parece por tanto tener mucho efecto a nivel génico el hecho de hacer ejercicio en ratones diabéticos. Por otro lado, al comparar ratones diabéticos con ejercicio *versus* sanos sí se ha observado una disminución en genes implicados en la formación de las fibras de colágeno.

Asimismo, abordando la cuestión de la variable tiempo que duró el experimento, el hecho de encontrar diferencias tal vez sea posible atribuirlos al envejecimiento de los ratones. Resulta imposible evaluarlo en este trabajo pues los ratones utilizados en todas las categorías tenían de 10 a 15 semanas de edad. Si se quisiera evaluar habría que utilizar ratones de la misma edad.

Finalmente, respecto a la parte metodológica, ha resultado tedioso y ha alargado el tiempo de entrega el hecho de encontrar errores con distintas librerías. En concreto, con la función `rma` de la librería `Affy`, y con las librerías `ReactomePA` y `reactome.db`. Con estas últimas, según la versión de R y la versión de éstas descargadas había ordenadores en los que el pipeline se podía ejecutar mientras que en otros con versiones más modernas no.

Conclusión

- Existen diferencias a nivel de expresión génica entre ratones sanos y diabéticos.
- El ejercicio tiene un efecto de reducción de la transcripción de genes implicados en la contracción muscular y conducción cardíaca en ratones sanos.
- Entre ratones diabéticos que han sido sometidos a un programa de ejercicio y aquellos que no, no se han observado diferencias notables a nivel de expresión génica.
- El ejercicio tiene un efecto en ratones diabéticos reduciendo la expresión de genes implicados en la formación de las fibras de colágeno.
-

Bibliografía

Baxter, R. V., Othmane, K. B., Rochelle, J. M., Stajich, J. E., Hulette, C., Dew-Knight, S., ... & Gilbert, J. R. (2002). Ganglioside-induced differentiation-associated protein-1 is mutant in Charcot-Marie-Tooth disease type 4A/8q21. *Nature genetics*, 30(1), 21-22.

Budanov, A. V., & Karin, M. (2008). p53 target genes sestrin1 and sestrin2 connect genotoxic stress and mTOR signaling. *Cell*, 134(3), 451-460.

Ching, Y. H., Ghosh, T. K., Cross, S. J., Packham, E. A., Honeyman, L., Loughna, S., ... & Thomas, N. R. (2005). Mutation in myosin heavy chain 6 causes atrial septal defect. *Nature genetics*, 37(4), 423-428.

Huang, X. P., & Du, J. F. (2004). Troponin I, cardiac diastolic dysfunction and restrictive cardiomyopathy. *Acta Pharmacologica Sinica*, 25, 1569-1575.

Lehti, T. M., Silvennoinen, M., Kivela, R., Kainulainen, H., & Komulainen, J. (2006). Effects of streptozotocin-induced diabetes and physical training on gene expression of extracellular matrix proteins in mouse skeletal muscle. *American Journal of Physiology-Endocrinology and Metabolism*, 290(5), E900-E907.

Lehti, T. M., Silvennoinen, M., Kivela, R., Kainulainen, H., & Komulainen, J. (2007). Effects of streptozotocin-induced diabetes and physical training on gene expression of titin-based stretch-sensing complexes in mouse striated muscle. *American Journal of Physiology-Endocrinology and Metabolism*, 292(2), E533-E542.

Kivela, R., Silvennoinen, M., Touva, A. M., Lehti, T. M., Kainulainen, H., Vihko, V., ... & Kainulainen, H. (2006). Effects of experimental type 1 diabetes and exercise training on angiogenic gene expression and capillarization in skeletal muscle. *The FASEB journal*, 20(9), 1570-1572.

Anexo: Código utilizado en R

PEC 1 A.D.O.

Sofía Zdral

16/4/2020

PEC 1: MICROARRAY ANALYSIS OF [GEO:GSE1659](#) “Time series of diabetes and exercise training induced expression changes in skeletal muscle of mice”

```
library("knitr")
library("colorspace")
library("gplots")

library("ggplot2")
library("ggrepel")
library("htmlTable")
library("prettydoc")
library("devtools")

library("BiocManager")

library("oligo")

library("arrayQualityMetrics")
library("pvca")

library("dplyr")

library("genefilter")
library("annotate")

library("org.Mm.eg.db")

library("ReactomePA")

library("reactome.db")
```

RESULTADOS:

PASO 1: Preparación de los datos

```
#setwd(".")
#setwd("D:/BIOESTADÍSTICA-INFORMÁTICA/Análisis de datos ómicos/data")
#dir.create("data")
#dir.create("results")
targets=read.csv(paste(getwd(),"/targets.csv", sep = ""),header=TRUE,sep=";") #Cargamos los datos del fichero targets
targets
```

##	i..FileName	Group	HealthStatus	Trained	Time
## 1	GSM28550.CEL	Healthy.NoTrained	Healthy	No	1_Week
## 2	GSM28551.CEL	Healthy.NoTrained	Healthy	No	3_Week
## 3	GSM28552.CEL	Healthy.NoTrained	Healthy	No	5_Week
## 4	GSM28553.CEL	Diabetic.NoTrained	Diabetic	No	1_Week
## 5	GSM28554.CEL	Diabetic.NoTrained	Diabetic	No	3_Week

```
## 6 GSM28555.CEL Diabetic.NoTrained Diabetic No 5_Week
## 7 GSM28556.CEL Diabetic.Trained Diabetic Yes 1_Week
## 8 GSM28557.CEL Diabetic.Trained Diabetic Yes 3_Week
## 9 GSM28558.CEL Diabetic.Trained Diabetic Yes 5_Week
## 10 GSM28559.CEL Healthy.Trained Healthy Yes 1_Week
## 11 GSM28560.CEL Healthy.Trained Healthy Yes 3_Week
## 12 GSM28561.CEL Healthy.Trained Healthy Yes 5_Week
##
## ShortName
## 1 Healthy_1W
## 2 Healthy_3W
## 3 Healthy_5W
## 4 Diabetic_1W
## 5 Diabetic_3W
## 6 Diabetic_5W
## 7 Trained_diabetic_1W
## 8 Trained_diabetic_3W
## 9 Trained_diabetic_5W
## 10 Trained_1W
## 11 Trained_3W
## 12 Trained_5W
```

`knitr::kable(targets, booktabs = TRUE, caption = 'Content of the targets file used for the current analysis')` #Ahora podemos ver en una tabla a todas las muestras de microarray junto a sus atributos. También tenemos la columna "ShortName", que contiene los nombres con los cuales podremos identificar cada muestra en todos los análisis que realicemos.

Content of the targets file used for the current analysis

File Name	Group	HealthStatus	Trained	Time	ShortName
GSM28550.CEL	Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
GSM28551.CEL	Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
GSM28552.CEL	Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
GSM28553.CEL	Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
GSM28554.CEL	Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
GSM28555.CEL	Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
GSM28556.CEL	Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
GSM28557.CEL	Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
GSM28558.CEL	Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W

GSM28559.CEL	Healthy.Trained	Healthy	Yes	1_Week	Trained_1W
GSM28560.CEL	Healthy.Trained	Healthy	Yes	3_Week	Trained_3W
GSM28561.CEL	Healthy.Trained	Healthy	Yes	5_Week	Trained_5W

Cargamos los ficheros .CEL y leemos los datos. Al final de este paso tendremos las intensidades "raw" guardadas en rawData.

```
library(oligo)
celFiles = list.celfiles(paste(getwd(), "/data", sep = ""), full.names = TRUE)
library(Biobase)
require(Biobase)
#Ahora vamos a asociar la información almacenada en los archivos CEL con el archivo "targets"
myTargets <- read.AnnotatedDataFrame(paste(getwd(), "/targets.csv", sep = ""), header = TRUE, row.names = 1, sep=";")
#Ahora tenemos los datos crudos "raw"
rawData = read.celfiles(celFiles, phenoData = myTargets)

## Loading required package: pd.mg.u74av2

## Loading required package: RSQLite

## Loading required package: DBI

## Platform design info loaded.

## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28550.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28551.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28552.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28553.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28554.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28555.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28556.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28557.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28558.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28559.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28560.CEL
## Reading in : C:/Users/monol/OneDrive/Escritorio/sofi/data/GSM28561.CEL
```

```
## Warning in read.celfiles(celFiles, phenoData = myTargets): 'channel
',
## automatically added to varMetadata in phenoData.

print(pData(rawData))

##              Group HealthStatus Trained   Time
ShortName
## GSM28550.CEL  Healthy.NoTrained   Healthy   No 1_Week
Healthy_1W
## GSM28551.CEL  Healthy.NoTrained   Healthy   No 3_Week
Healthy_3W
## GSM28552.CEL  Healthy.NoTrained   Healthy   No 5_Week
Healthy_5W
## GSM28553.CEL  Diabetic.NoTrained  Diabetic   No 1_Week
Diabetic_1W
## GSM28554.CEL  Diabetic.NoTrained  Diabetic   No 3_Week
Diabetic_3W
## GSM28555.CEL  Diabetic.NoTrained  Diabetic   No 5_Week
Diabetic_5W
## GSM28556.CEL  Diabetic.Trained     Diabetic   Yes 1_Week Trained
_diabetic_1W
## GSM28557.CEL  Diabetic.Trained     Diabetic   Yes 3_Week Trained
_diabetic_3W
## GSM28558.CEL  Diabetic.Trained     Diabetic   Yes 5_Week Trained
_diabetic_5W
## GSM28559.CEL  Healthy.Trained      Healthy    Yes 1_Week
Trained_1W
## GSM28560.CEL  Healthy.Trained      Healthy    Yes 3_Week
Trained_3W
## GSM28561.CEL  Healthy.Trained      Healthy    Yes 5_Week
Trained_5W

#Nombramos las muestras con el ShortName
colnames(rawData) = myTargets@data$ShortName
head(rawData)

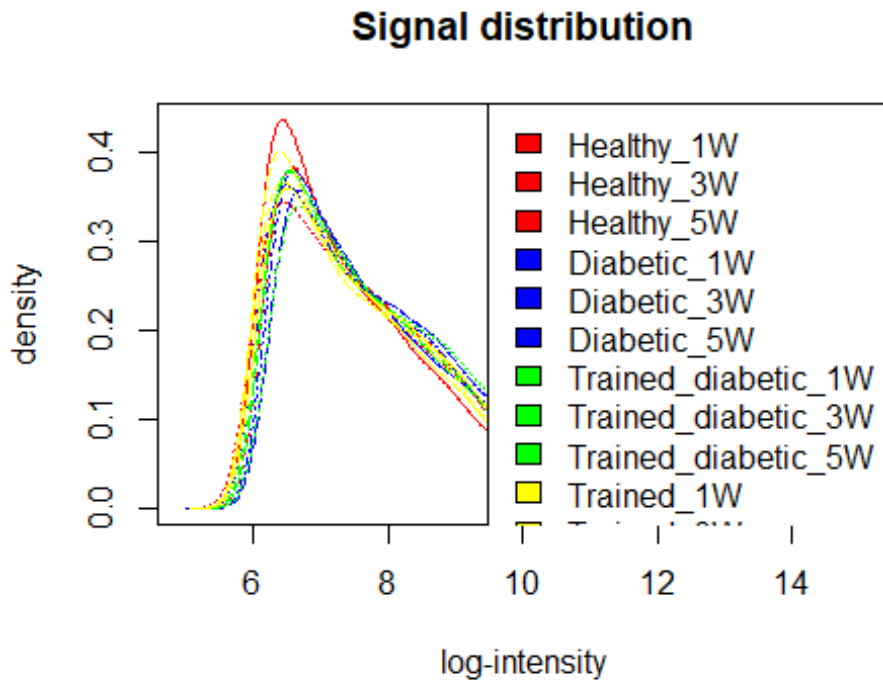
## ExpressionFeatureSet (storageMode: lockedEnvironment)
## assayData: 6 features, 12 samples
##   element names: exprs
## protocolData
##   rowNames: Healthy_1W Healthy_3W ... Trained_5W (12 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: Healthy_1W Healthy_3W ... Trained_5W (12 total)
##   varLabels: Group HealthStatus ... ShortName (5 total)
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.mg.u74av2
```

PASO 2: Control de calidad

Histograma con rawData

```
ShortName<-rawData$ShortName
```

```
hist(rawData, main = "Signal distribution", col = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)))
legend (x="topright", legend=ShortName, fill = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)))
```



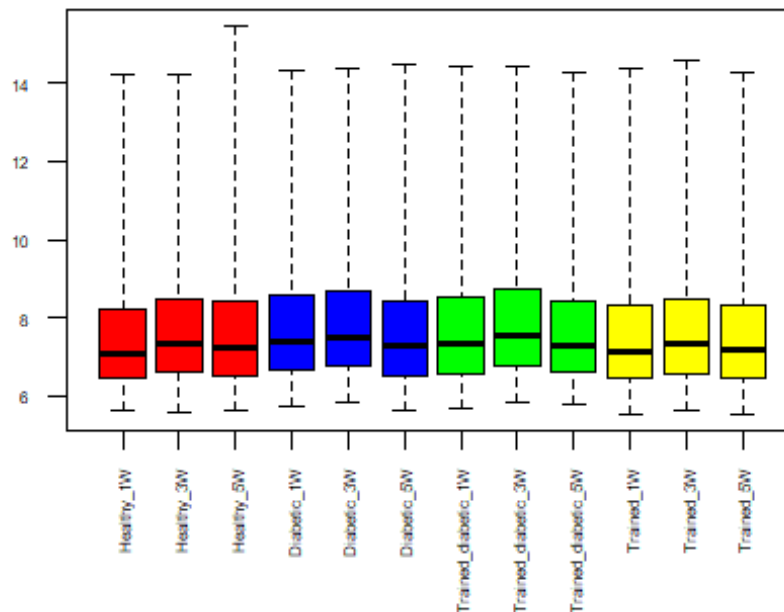
```
png("hist_RawData.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
hist(rawData, main = "Signal distribution", col = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)))
legend (x="topright", legend=ShortName, fill = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)))
dev.off()
```

```
## png
## 2
```

Diagrama de cajas y bigotes con rawData

```
boxplot(rawData, cex.axis=0.5, las=2, which="all",
col = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)),
main="Distribution of raw intensity values")
```

Distribution of raw intensity values



```
png("boxplot_RawData.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
boxplot(rawData, cex.axis=0.5, las=2, which="all",
col = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow",
3)),
main="Distribution of raw intensity values")
dev.off()
```

```
## png
## 2
```

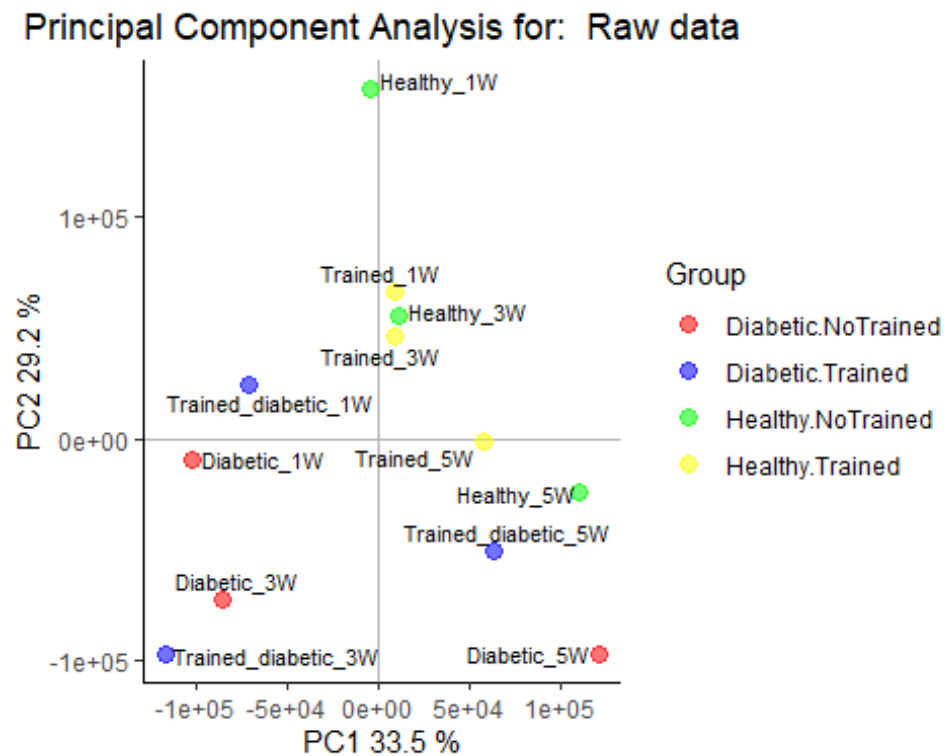
Análisis de Componentes Principales (PCA) con rawData

```
library(ggplot2)
library(ggrepel)
plotPCA3 <- function (datos, labels, factor, title, scale,colores, size = 1.5, glineas = 0.25) {
  data <- prcomp(t(datos),scale=scale)
  # plot adjustments
  dataDf <- data.frame(data$x)
  Group <- factor
  loads <- round(data$sdev^2/sum(data$sdev^2)*100,1)
  # main plot
  p1 <- ggplot(dataDf,aes(x=PC1, y=PC2)) +
  theme_classic() +
  geom_hline(yintercept = 0, color = "gray70") +
  geom_vline(xintercept = 0, color = "gray70") +
  geom_point(aes(color = Group), alpha = 0.55, size = 3) +
  coord_cartesian(xlim = c(min(data$x[,1])-5,max(data$x[,1])+5)) +
  scale_fill_discrete(name = "Group")
  # avoiding labels superposition
  p1 + geom_text_repel(aes(y = PC2 + 0.25, label = labels),segment.size
```

```

= 0.25, size = size) +
labs(x = c(paste("PC1",loads[1],"%")),y=c(paste("PC2",loads[2],"%")))
+ ggtitle(paste("Principal Component Analysis for: ",title,sep=" ")) +
theme(plot.title = element_text(hjust = 0.5)) +
scale_color_manual(values=colores)
}
plotPCA3(exprs(rawData), labels = targets$ShortName, factor = targets$
Group, title="Raw data", scale = FALSE, size = 3, colores = c("red",
"blue","green","yellow"))

```



```

png("PCA_RawData.png", width = 20, height = 12,
units = "cm", res = 600, pointsize = 10)
plotPCA3 <- function (datos, labels, factor, title, scale,colores, siz
e = 1.5, glineas = 0.25) {
data <- prcomp(t(datos),scale=scale)
# plot adjustments
dataDf <- data.frame(data$x)
Group <- factor
loads <- round(data$sdev^2/sum(data$sdev^2)*100,1)
# main plot
p1 <- ggplot(dataDf,aes(x=PC1, y=PC2)) +
theme_classic() +
geom_hline(yintercept = 0, color = "gray70") +
geom_vline(xintercept = 0, color = "gray70") +
geom_point(aes(color = Group), alpha = 0.55, size = 3) +
coord_cartesian(xlim = c(min(data$x[,1])-5,max(data$x[,1])+5)) +
scale_fill_discrete(name = "Group")
# avoiding labels superposition
p1 + geom_text_repel(aes(y = PC2 + 0.25, label = labels),segment.size
= 0.25, size = size) +
labs(x = c(paste("PC1",loads[1],"%")),y=c(paste("PC2",loads[2],"%")))

```

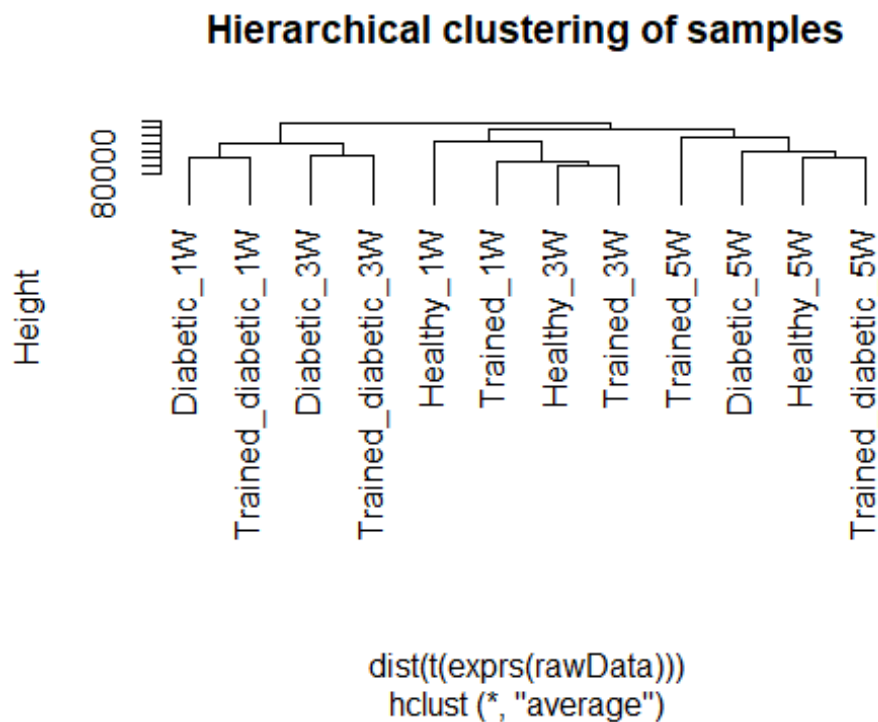


```
+ ggtitle(paste("Principal Component Analysis for: ",title,sep=" ")) +
theme(plot.title = element_text(hjust = 0.5)) +
scale_color_manual(values=colores)
}
plotPCA3(exprs(rawData), labels = targets$ShortName, factor = targets$
Group, title="Raw data", scale = FALSE, size = 3, colores = c("red",
"blue","green","yellow"))
dev.off()

## png
## 2
```

Clúster jerárquico con rawData

```
hc <- hclust(dist(t(exprs(rawData))),method="average")
plot(hc, labels=ShortName, hang=-1, main="Hierarchical clustering of s
amples")
```



```
png("Cluster_RawData.png", width = 20, height = 12,
units = "cm", res = 600, pointsize = 10)
plot(hc, labels=ShortName, hang=-1, main="Hierarchical clustering of s
amples")
dev.off()

## png
## 2
```

Finalmente, utilizamos la librería “arrayQualityMetrics” para generar un informe de la calidad de nuestras muestras. Este informe llamado “index.html” se ha generado en la carpeta Results/QCDir.Norm

```
library(arrayQualityMetrics)
arrayQualityMetrics(rawData, outdir = "./Results/rawData_quality", force = T)
```

```
## The report will be written into directory './Results/rawData_quality'.
```

```
#!/[Caption for the picture.](/C:/Users/monol/OneDrive/Escritorio/sofi/Muestras_array_raw_cruces.png)
```

array	sampleNames	*1	*2	*3	Group	HealthStatus	Trained	Time	ShortName
<input type="checkbox"/>	1	Healthy_1W			Healthy.NoTrained	Healthy	No	1_Week	Healthy_1W
<input type="checkbox"/>	2	Healthy_3W			Healthy.NoTrained	Healthy	No	3_Week	Healthy_3W
<input type="checkbox"/>	3	Healthy_5W			Healthy.NoTrained	Healthy	No	5_Week	Healthy_5W
<input type="checkbox"/>	4	Diabetic_1W			Diabetic.NoTrained	Diabetic	No	1_Week	Diabetic_1W
<input type="checkbox"/>	5	Diabetic_3W			Diabetic.NoTrained	Diabetic	No	3_Week	Diabetic_3W
<input type="checkbox"/>	6	Diabetic_5W	x		Diabetic.NoTrained	Diabetic	No	5_Week	Diabetic_5W
<input type="checkbox"/>	7	Trained_diabetic_1W			Diabetic.Trained	Diabetic	Yes	1_Week	Trained_diabetic_1W
<input type="checkbox"/>	8	Trained_diabetic_3W			Diabetic.Trained	Diabetic	Yes	3_Week	Trained_diabetic_3W
<input type="checkbox"/>	9	Trained_diabetic_5W			Diabetic.Trained	Diabetic	Yes	5_Week	Trained_diabetic_5W
<input type="checkbox"/>	10	Trained_1W			Healthy.Trained	Healthy	Yes	1_Week	Trained_1W
<input type="checkbox"/>	11	Trained_3W			Healthy.Trained	Healthy	Yes	3_Week	Trained_3W
<input type="checkbox"/>	12	Trained_5W			Healthy.Trained	Healthy	Yes	5_Week	Trained_5W

Análisis de calidad mediante 3 métodos de los datos de microarray crudos “raw”

PASO 3: Normalización y control de calidad de datos normalizados

Normalización de las muestras por el método RMA

```
eset_rma<-rma(rawData)
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

Sacamos de nuevo el informe de calidad pero con nuestras muestras normalizadas.

```
arrayQualityMetrics(eset_rma, outdir = file.path("./results", "QCDir.Norm"), force=TRUE)
```

```
## The report will be written into directory './results/QCDir.Norm'.
```

```
#!/[Caption for the picture.](/C:/Users/monol/OneDrive/Escritorio/sofi/Análisis de datos ómicos/Muestras_array_norm_cruces.png)
```

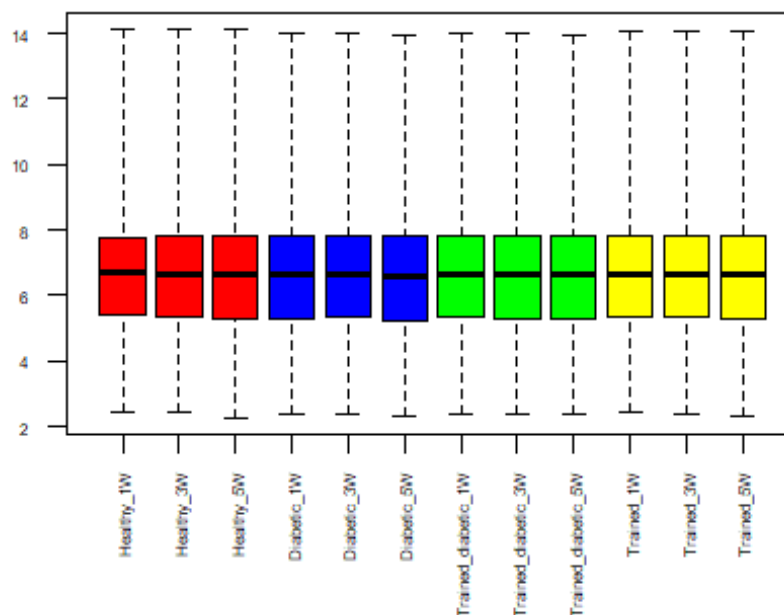
array	sampleNames	*1 *2 *3	Group	HealthStatus	Trained	Time	ShortName
<input type="checkbox"/>	1	Healthy_1W	x	Healthy.NoTrained	Healthy	No 1_Week	Healthy_1W
<input type="checkbox"/>	2	Healthy_3W		Healthy.NoTrained	Healthy	No 3_Week	Healthy_3W
<input type="checkbox"/>	3	Healthy_5W		Healthy.NoTrained	Healthy	No 5_Week	Healthy_5W
<input type="checkbox"/>	4	Diabetic_1W		Diabetic.NoTrained	Diabetic	No 1_Week	Diabetic_1W
<input type="checkbox"/>	5	Diabetic_3W	x	Diabetic.NoTrained	Diabetic	No 3_Week	Diabetic_3W
<input type="checkbox"/>	6	Diabetic_5W		Diabetic.NoTrained	Diabetic	No 5_Week	Diabetic_5W
<input type="checkbox"/>	7	Trained_diabetic_1W		Diabetic.Trained	Diabetic	Yes 1_Week	Trained_diabetic_1W
<input type="checkbox"/>	8	Trained_diabetic_3W	x	Diabetic.Trained	Diabetic	Yes 3_Week	Trained_diabetic_3W
<input type="checkbox"/>	9	Trained_diabetic_5W		Diabetic.Trained	Diabetic	Yes 5_Week	Trained_diabetic_5W
<input type="checkbox"/>	10	Trained_1W		Healthy.Trained	Healthy	Yes 1_Week	Trained_1W
<input type="checkbox"/>	11	Trained_3W		Healthy.Trained	Healthy	Yes 3_Week	Trained_3W
<input type="checkbox"/>	12	Trained_5W		Healthy.Trained	Healthy	Yes 5_Week	Trained_5W

Análisis de calidad mediante 3 métodos de los datos de microarray normalizados

Diagrama de cajas y bigotes con los datos normalizados

```
boxplot(eset_rma, cex.axis=0.5, las=2, which="all", col = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)), main="Boxplot for arrays intensity: Normalized Data")
```

Boxplot for arrays intensity: Normalized Data



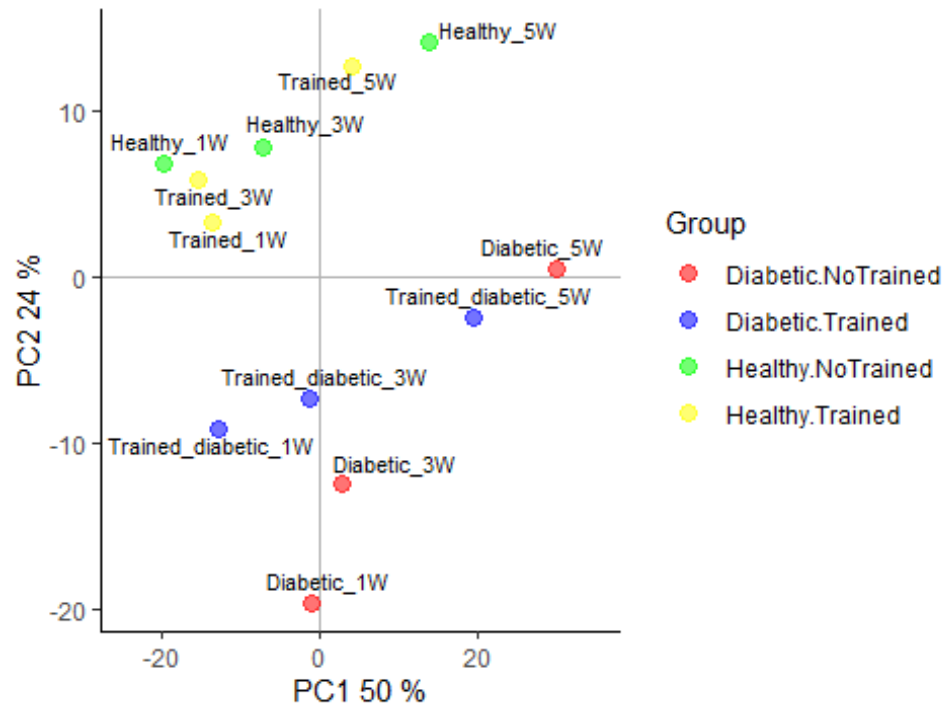
```
png("Boxplot_normData.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
boxplot(eset_rma, cex.axis=0.5, las=2, which="all", col = c(rep("red", 3), rep("blue", 3), rep("green", 3), rep("yellow", 3)), main="Boxplot for arrays intensity: Normalized Data")
dev.off()

## png
## 2
```

Análisis de componentes principales con los datos normalizados

```
plotPCA3(exprs(eset_rma), labels = targets$ShortName, factor = targets$Group, title="Normalized data", scale = FALSE, size = 3, colores = c("red", "blue", "green", "yellow"))
```

Principal Component Analysis for: Normalized data



```
png("PCA_NormData.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
PCA_NormData<-plotPCA3(exprs(eset_rma), labels = targets$ShortName, factor = targets$Group, title="Normalized data", scale = FALSE, size = 3, colores = c("red", "blue", "green", "yellow"))
PCA_NormData
dev.off()

## png
## 2
```

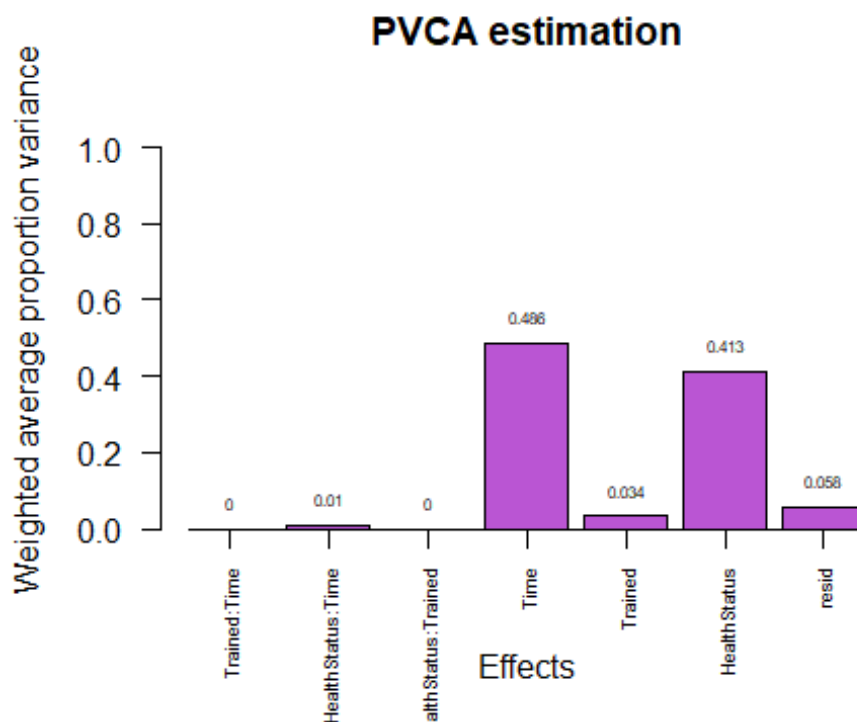
PASO 4: Detección de efectos derivados del "batch"

Principal Variance Component Analysis (PVCA) para buscar efectos producidos por el "batch"

```
library(pvca)
pData(eset_rma) <- targets
#select the threshold
pct_threshold <- 0.6
#select the factors to analyze
batch.factors <- c("HealthStatus", "Trained","Time")
#run the analysis
pvcaObj <- pvcaBatchAssess (eset_rma, batch.factors, pct_threshold)
```

```
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular

bp <- barplot(pvcaObj$dat, xlab = "Effects",
  ylab = "Weighted average proportion variance",
  ylim= c(0,1.1),col = c("mediumorchid"), las=2,
  main="PVCA estimation")
axis(1, at = bp, labels = pvcaObj$label, cex.axis = 0.55, las=2)
values = pvcaObj$dat
new_values = round(values , 3)
text(bp,pvcaObj$dat,labels = new_values, pos=3, cex = 0.5)
```



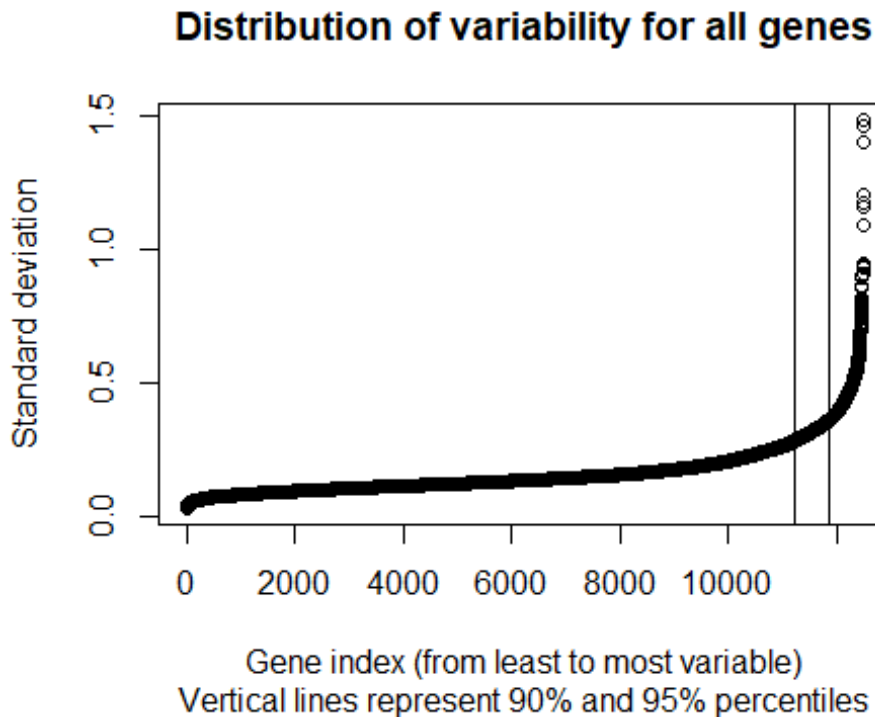
```
png("PVCA_estimation.png", width = 20, height = 12,
  units = "cm", res = 600, pointsize = 10)
bp <- barplot(pvcaObj$dat, xlab = "Effects",
  ylab = "Weighted average proportion variance",
  ylim= c(0,1.1),col = c("mediumorchid"), las=2,
  main="PVCA estimation")
axis(1, at = bp, labels = pvcaObj$label, cex.axis = 0.55, las=2)
values = pvcaObj$dat
new_values = round(values , 3)
text(bp,pvcaObj$dat,labels = new_values, pos=3, cex = 0.5)
dev.off()

## png
## 2
```

PASO 5: Detección de los genes más y menos variables y filtraje

Genes más variables entre muestras con una desviación estándar superior al 90-95%.

```
sds <- apply (exprs(eset_rma), 1, sd)
sds0<- sort(sds)
plot(1:length(sds0), sds0, main="Distribution of variability for all g
enes", sub="Vertical lines represent 90% and 95% percentiles", xlab="G
ene index (from least to most variable)", ylab="Standard deviation")
abline(v=length(sds)*c(0.9,0.95))
```



```
png("Var_genes.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
plot(1:length(sds0), sds0, main="Distribution of variability for all g
enes", sub="Vertical lines represent 90% and 95% percentiles", xlab="G
ene index (from least to most variable)", ylab="Standard deviation")
abline(v=length(sds)*c(0.9,0.95))
dev.off()

## png
## 2
```

Filtraje de aquellos genes cuya variabilidad es posible atribuirla a la variación aleatoria más que a diferencias entre situaciones de nuestro experimento. El umbral de variabilidad es de 0.75. Anotación de los datos de microarray: "mgu74av2.db".

```
library(genefilter)
library(mgu74av2.db)

##

annotation(eset_rma) <- "mgu74av2.db"
filtered <- nsFilter(eset_rma, require.entrez = TRUE, remove.dupEntrez
```

```
= TRUE, var.filter=TRUE, var.func=IQR, var.cutoff=0.75, filterByQuantile=TRUE, feature.exclude = "^AFFX")
print(filtered$filter.log)
```

```
## $numDupsRemoved
## [1] 2639
##
## $numLowVar
## [1] 6526
##
## $numRemoved.ENTREZID
## [1] 1125
##
## $feature.exclude
## [1] 22
```

```
eset_filtered <- filtered$eset
```

Nos hemos quedado con 2176 genes después del filtrado para analizar. Ahora guardamos los datos normalizados y filtrados:

```
library("readr")
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:genefilter':
##
##      spec
```

```
write.csv(exprs(eset_rma), file="./results/normalized.Data.csv")
```

```
write.csv(exprs(eset_filtered), file="./results/normalized.Filtered.Data.csv")
```

```
save(eset_rma, eset_filtered, file="./results/normalized.Data.Rda")
```

PASO 6: Matriz de diseño y matriz de contraste

```
if (!exists("eset_filtered")) load (file="./results/normalized.Data.Rda")
```

Matriz de diseño: 12 filas (1 por cada muestra) y 4 columnas ya que hemos considerado nuestro experimento como un experimento con un solo factor y cuatro niveles.

```
library(limma)
```

```
designMat<- model.matrix(~0+Group, pData(eset_filtered))
```

```
colnames(designMat) <- c("Diabetic.NoTrained", "Diabetic.Trained", "Healthy.NoTrained", "Healthy.Trained")
```

```
print(designMat)
```

```
##      Diabetic.NoTrained Diabetic.Trained Healthy.NoTrained Healthy.Trained
## 1                      0                0                  1
## 2                      0                0                  1
## 3                      0                0                  1
```



```
## 4      1      0      0
0
## 5      1      0      0
0
## 6      1      0      0
0
## 7      0      1      0
0
## 8      0      1      0
0
## 9      0      1      0
0
## 10     0      0      0
1
## 11     0      0      0
1
## 12     0      0      0
1
## attr("assign")
## [1] 1 1 1 1
## attr("contrasts")
## attr("contrasts")$Group
## [1] "contr.treatment"
```

Matriz de contrastes. Teniendo en cuenta los objetivos de este trabajo, los contrastes que se realizarán son los siguientes: (1) Ratón sano vs sano con entrenamiento, (2) Ratón diabético vs ratón sano, (3) Ratón diabético con entrenamiento vs ratón sano, (4) Ratón diabético entrenado vs ratón diabético y (5) La interacción entre estado de salud y entrenamiento ("INT").

```
cont.matrix <- makeContrasts (TrainedvsNoTrained.Healthy = Healthy.Trained-Healthy.NoTrained, DiabeticvsControl.NoTrained = Diabetic.NoTrained-Healthy.NoTrained, DiabeticTrainedvsControl = Diabetic.Trained-Healthy.NoTrained, TrainedcvvsNoTrained.Diabetic = Diabetic.Trained-Diabetic.NoTrained, INT = (Diabetic.Trained-Healthy.Trained) - (Diabetic.NoTrained-Healthy.NoTrained), levels=designMat)
print(cont.matrix)
```

```
##              Contrasts
## Levels      TrainedvsNoTrained.Healthy DiabeticvsControl.NoTrained
## Diabetic.NoTrained      0
1
## Diabetic.Trained      0
0
## Healthy.NoTrained     -1
-1
## Healthy.Trained      1
0
##              Contrasts
## Levels      DiabeticTrainedvsControl TrainedcvvsNoTrained.Diabetic INT
## Diabetic.NoTrained      0
-1 -1
## Diabetic.Trained      1
```

```

1 1
## Healthy.NoTrained -1
0 1
## Healthy.Trained 0
0 -1

```

3.7. Selección de genes diferencialmente expresados

Se utilizarán modelos de Bayes empíricos para combinar la información de la matriz y los genes y mejorar así las estimaciones del error. Asimismo, se utilizará el método de Benjamini-Hochberg para ajustar los p-valores obtenidos de forma que se pueda controlar la tasa de falsos positivos.

```

library(limma)
fit<-lmFit(eset_filtered, designMat)
fit.main<-contrasts.fit(fit, cont.matrix)
fit.main<-eBayes(fit.main)
class(fit.main)

## [1] "MAarrayLM"
## attr(,"package")
## [1] "limma"

```

Genes diferencialmente expresados en cada una de las comparaciones ordenados de más a menos diferencialmente expresados. Se mostrarán los 5 genes más diferencialmente expresados con sus correspondientes valores de logFoldChange (logFC), expresión media (AveExpr), valor t, p-valor, p-valor ajustado, valor B:

Comparación 1: Ratón sano vs sano con entrenamiento

```

topHealthy.TrainedvsHealthy.NoTrained <- topTable (fit.main, number=nrow(fit.main), coef="TrainedvsNoTrained.Healthy", adjust="fdr")

```

```

head(topHealthy.TrainedvsHealthy.NoTrained, 5) #Genes más diferencialmente expresados

```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
## 100921_at	-1.6954610	6.947136	-7.828799	3.413894e-06	0.007428633	-0.2858227
## 101071_at	-1.8109932	5.483442	-5.635183	9.083567e-05	0.098829213	-1.1743220
## 98569_at	-0.5198317	9.296581	-2.932546	1.197232e-02	0.999521589	-3.1368634
## 100593_at	-0.8750638	5.957940	-2.891331	1.294597e-02	0.999521589	-3.1741444
## 160901_at	-0.7890120	6.087163	-2.805927	1.521973e-02	0.999521589	-3.2517406

Comparación 2: Ratón diabético vs sano

```

topDiabetic.NoTrainedvsHealthy.NoTrained <- topTable (fit.main, number=nrow(fit.main), coef="DiabeticvsControl.NoTrained", adjust="fdr")

```

```

head(topDiabetic.NoTrainedvsHealthy.NoTrained, 5) #Genes más diferencialmente expresados

```

```
##          logFC AveExpr      t      P.Value    adj.P.Val
B
## 95731_at    2.322352 8.008379 12.456348 1.845874e-08 4.016622e-05 9.
388157
## 160522_at -1.536395 8.954212 -9.457259 4.351244e-07 4.734154e-04 6.
700336
## 94297_at    1.984398 9.089250  8.345269 1.721765e-06 1.248854e-03 5.
456739
## 100921_at -1.740535 6.947136 -8.036927 2.581357e-06 1.404258e-03 5.
083766
## 102967_at -1.929720 6.803440 -7.185601 8.370910e-06 3.491158e-03 3.
985178
```

Comparación 3: Ratón diabético con entrenamiento vs ratón sano

```
topDiabetic.TrainedvsHealthy.NoTrained <- topTable (fit.main, number=n
row(fit.main), coef="DiabeticTrainedvsControl", adjust="fdr")
```

head(topDiabetic.TrainedvsHealthy.NoTrained, 5) *#Genes más diferencialmente expresados*

```
##          logFC AveExpr      t      P.Value    adj.P.Val
B
## 95731_at    1.673315 8.008379  8.975122 7.776875e-07 0.001692248 6.0
14329
## 100921_at -1.640036 6.947136 -7.572874 4.848379e-06 0.003233014 4.4
13217
## 94297_at    1.768267 9.089250  7.436343 5.865241e-06 0.003233014 4.2
42516
## 160522_at -1.206558 8.954212 -7.426954 5.943041e-06 0.003233014 4.2
30676
## 96221_at    1.049813 7.175732  6.748565 1.586386e-05 0.006903950 3.3
39623
```

Comparación 4: Ratón diabético entrenado vs ratón diabético

```
topDiabetic.TrainedvsDiabetic.NoTrained <- topTable (fit.main, number=
nrow(fit.main), coef="TrainedcvvsNoTrained.Diabetic", adjust="fdr")
```

head(topDiabetic.TrainedvsDiabetic.NoTrained, 5)

```
##          logFC AveExpr      t      P.Value    adj.P.Val
B
## 162015_f_at  0.8177101 9.547900  3.599808 0.003378752 0.9988969 -3
.918268
## 95731_at    -0.6490371 8.008379 -3.481226 0.004226301 0.9988969 -3
.946285
## 103084_at   -0.9001082 10.238450 -3.442440 0.004548028 0.9988969 -3
.955625
## 98083_at    -0.5529877 7.725916 -3.345733 0.005462474 0.9988969 -3
.979287
## 94534_at     0.4813928 9.563225  3.289206 0.006080818 0.9988969 -3
.993362
```

Comparación 5: Interacción

```
topTab_INT <- topTable (fit.main, number=nrow(fit.main), coef="INT",
adjust="fdr")
```

```
head(topTab_INT, 5)
```

```
##           logFC  AveExpr      t      P.Value adj.P.Val
B
## 100921_at  1.7959597  6.947136  5.863932 6.258452e-05 0.1361839 -3.
031140
## 101071_at  1.5922214  5.483442  3.503319 4.053454e-03 0.9988892 -3.
652202
## 98569_at   0.6760587  9.296581  2.696817 1.870404e-02 0.9988892 -3.
962202
## 103084_at -0.9226285 10.238450 -2.495075 2.731332e-02 0.9988892 -4.
045538
## 94817_at   0.5941550  7.538252  2.421032 3.135062e-02 0.9988892 -4.
076451
```

Como podemos observar, la primera columna de las tablas obtenidas en las cinco comparaciones corresponde al ID de cada conjunto de sondas de Affymetrix. Para poder conocer los genes diferencialmente expresados necesitamos conocer qué gen corresponde a cada ID (anotación).

```
annotatedTopTable <- function(topTab, anotPackage)
{
  topTab <- cbind(PROBEID=rownames(topTab), topTab)
  myProbes <- rownames(topTab)
  thePackage <- eval(parse(text = anotPackage))
  geneAnots <- select(thePackage, myProbes, c("SYMBOL", "ENTREZID", "GEN
ENAME"))
  annotatedTopTab<- merge(x=geneAnots, y=topTab, by.x="PROBEID", by.y="P
ROBEID")
  return(annotatedTopTab)
}

topAnnotated_HealthyTrainedvsHealthy.NoTrained<- annotatedTopTable(top
Healthy.TrainedvsHealthy.NoTrained, anotPackage="mgu74av2.db")

## 'select()' returned 1:1 mapping between keys and columns

topAnnotated_Diabetic.NoTrainedvsHealthy.NoTrained<- annotatedTopTable
(topDiabetic.NoTrainedvsHealthy.NoTrained, anotPackage="mgu74av2.db")

## 'select()' returned 1:1 mapping between keys and columns

topAnnotated_Diabetic.TrainedvsHealthy.NoTrained<- annotatedTopTable(t
opDiabetic.TrainedvsHealthy.NoTrained, anotPackage="mgu74av2.db")

## 'select()' returned 1:1 mapping between keys and columns

topAnnotated_Diabetic.TrainedvsDiabetic.NoTrained<- annotatedTopTable(
topDiabetic.TrainedvsDiabetic.NoTrained, anotPackage="mgu74av2.db")

## 'select()' returned 1:1 mapping between keys and columns

topAnnotated_INT<- annotatedTopTable(topTab_INT, anotPackage="mgu74av2
.db")

## 'select()' returned 1:1 mapping between keys and columns
```

```

write.csv(topAnnotated_HealthyTrainedvsHealthy.NoTrained, file="./results/topAnnotated_HealthyTrainedvsHealthy.NoTrained.csv")
write.csv(topAnnotated_Diabetic.NoTrainedvsHealthy.NoTrained, file="./results/topAnnotated_Diabetic.NoTrainedvsHealthy.NoTrained.csv")
write.csv(topAnnotated_Diabetic.TrainedvsHealthy.NoTrained, file="./results/topAnnotated_Diabetic.TrainedvsHealthy.NoTrained.csv")
write.csv(topAnnotated_Diabetic.TrainedvsDiabetic.NoTrained, file="./results/topAnnotated_Diabetic.TrainedvsDiabetic.NoTrained.csv")
write.csv(topAnnotated_INT, file="./results/topAnnotated_INT.csv")

```

```
head(topAnnotated_HealthyTrainedvsHealthy.NoTrained)
```

##	PROBEID	SYMBOL	ENTREZID	GENE
##				
## 1	100011_at	Klf3	16599	Kruppel-like factor 3 (basic)
## 2	100017_at	Mybph	53311	myosin binding protein H
## 3	100022_at	Cish	12700	cytokine inducible SH2-containing protein
## 4	100032_at	Sp1	20683	trans-acting transcription factor 1
## 5	100037_at	Ddx18	66942	DEAD (Asp-Glu-Ala-Asp) box polypeptide 18
## 6	100041_at	Slc25a39	68066	solute carrier family 25, member 39
##		logFC	AveExpr	t P.Value adj.P.Val B
## 1	0.02931764	5.044166	0.1645098	0.8719351 0.9995216 -5.054431
## 2	0.08898649	7.765299	0.3336887	0.7440930 0.9995216 -5.026991
## 3	-0.24595596	8.434054	-1.5360801	0.1491992 0.9995216 -4.377779
## 4	0.21557721	6.853628	0.5961384	0.5616217 0.9995216 -4.948950
## 5	0.07154840	5.823712	0.4226695	0.6796480 0.9995216 -5.005261
## 6	0.17715972	8.540386	1.3362499	0.2050588 0.9995216 -4.529766

```
head(topAnnotated_Diabetic.NoTrainedvsHealthy.NoTrained)
```

##	PROBEID	SYMBOL	ENTREZID	GENE
##				
## 1	100011_at	Klf3	16599	Kruppel-like factor 3 (basic)
## 2	100017_at	Mybph	53311	myosin binding protein H
## 3	100022_at	Cish	12700	cytokine inducible SH2-containing protein
## 4	100032_at	Sp1	20683	trans-acting transcription factor 1
## 5	100037_at	Ddx18	66942	DEAD (Asp-Glu-Ala-Asp) box polypeptide 18
## 6	100041_at	Slc25a39	68066	solute carrier family 25, member 39
##		logFC	AveExpr	t P.Value adj.P.Val B
## 1	-0.09394273	5.044166	-0.5271399	0.6072376553 0.81750081 -6.3093190
## 2	-1.19830110	7.765299	-4.4934859	0.0006494488 0.01745569 -0.206411

```

7
## 3 -0.39040700 8.434054 -2.4382268 0.0303647206 0.12396554 -3.902821
0
## 4 0.11856917 6.853628 0.3278808 0.7483767746 0.88954902 -6.397640
0
## 5 0.51400755 5.823712 3.0364803 0.0098279740 0.06662203 -2.838605
2
## 6 -0.39495679 8.540386 -2.9790124 0.0109614896 0.06973252 -2.942812
3

```

```
head(topAnnotated_Diabetic.TrainedvsHealthy.NoTrained)
```

##	PROBEID	SYMBOL	ENTREZID	GENE
## 1	100011_at	Klf3	16599	Kruppel-like factor 3 (basic)
## 2	100017_at	Mybph	53311	myosin binding protein H
## 3	100022_at	Cish	12700	cytokine inducible SH2-containing protein
## 4	100032_at	Sp1	20683	trans-acting transcription factor 1
## 5	100037_at	Ddx18	66942	DEAD (Asp-Glu-Ala-Asp) box polypeptide 18
## 6	100041_at	Slc25a39	68066	solute carrier family 25, member 39

##	logFC	AveExpr	t	P.Value	adj.P.Val	B
## 1	-0.07431921	5.044166	-0.41702656	0.683664314	0.97997666	-6.211600
## 2	-1.10572380	7.765299	-4.14633207	0.001220822	0.07379191	-0.746282
## 3	-0.40333688	8.434054	-2.51897833	0.026120651	0.26007869	-3.639628
## 4	0.02183925	6.853628	0.06039236	0.952788880	0.99930078	-6.299811
## 5	0.41471517	5.823712	2.44991424	0.029711704	0.26074342	-3.757985
## 6	-0.07242298	8.540386	-0.54625962	0.594411989	0.95952558	-6.147869

```
head(topAnnotated_Diabetic.TrainedvsDiabetic.NoTrained)
```

##	PROBEID	SYMBOL	ENTREZID	GENE
## 1	100011_at	Klf3	16599	Kruppel-like factor 3 (basic)
## 2	100017_at	Mybph	53311	myosin binding protein H
## 3	100022_at	Cish	12700	cytokine inducible SH2-containing protein
## 4	100032_at	Sp1	20683	trans-acting transcription factor 1
## 5	100037_at	Ddx18	66942	DEAD (Asp-Glu-Ala-Asp) box polypeptide 18
## 6	100041_at	Slc25a39	68066	solute carrier family 25, member 39

##	logFC	AveExpr	t	P.Value	adj.P.Val	B
## 1	0.01962353	5.044166	0.11011329	0.9140510	0.9988969	-4.717185
## 2	0.09257730	7.765299	0.34715382	0.7341956	0.9988969	-4.704504
## 3	-0.01292988	8.434054	-0.08075155	0.9369059	0.9988969	-4.717846
## 4	-0.09672992	6.853628	-0.26748848	0.7934140	0.9988969	-4.710209

```
## 5 -0.09929238 5.823712 -0.58656602 0.5678380 0.9988969 -4.678947
## 6 0.32253381 8.540386 2.43275273 0.0306753 0.9988969 -4.225357
```

```
head(topAnnotated_INT)
```

```
##      PROBEID  SYMBOL ENTREZID      GENE
NAME
## 1 100011_at    Klf3    16599      Kruppel-like factor 3 (ba
sic)
## 2 100017_at    Mybph    53311      myosin binding prote
in H
## 3 100022_at    Cish    12700 cytokine inducible SH2-containing pro
tein
## 4 100032_at    Sp1    20683      trans-acting transcription fact
or 1
## 5 100037_at    Ddx18    66942 DEAD (Asp-Glu-Ala-Asp) box polypeptid
e 18
## 6 100041_at Slc25a39    68066      solute carrier family 25, membe
r 39
##      logFC AveExpr      t  P.Value adj.P.Val      B
## 1 -0.009694115 5.044166 -0.038464124 0.9699193 0.9988892 -4.780143
## 2 0.003590816 7.765299 0.009521285 0.9925521 0.9988892 -4.780375
## 3 0.233026088 8.434054 1.029072667 0.3227627 0.9988892 -4.615156
## 4 -0.312307121 6.853628 -0.610676448 0.5522519 0.9988892 -4.719597
## 5 -0.170840782 5.823712 -0.713637262 0.4884378 0.9988892 -4.698102
## 6 0.145374095 8.540386 0.775344567 0.4524184 0.9988892 -4.683830
```

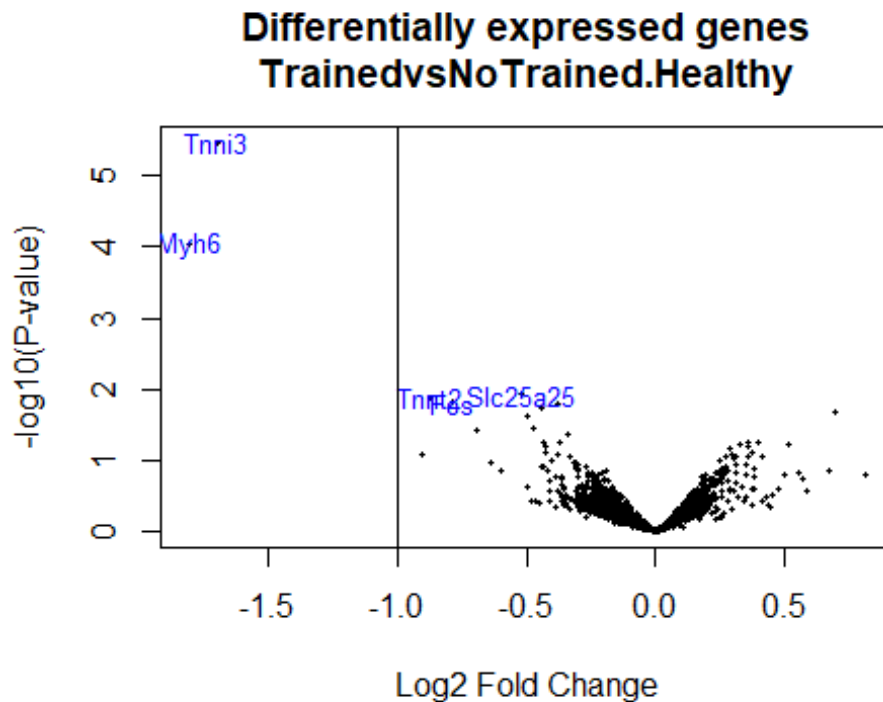
Volcano plots

Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 1: Ratón sano vs sano con entrenamiento.

```
library("mgu74av2.db")
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=1, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[1],
sep="\n"))
abline(v=c(-1,1))
```



```
png("Volcano_c1.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=1, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[1],
sep="\n"))
abline(v=c(-1,1))
dev.off()

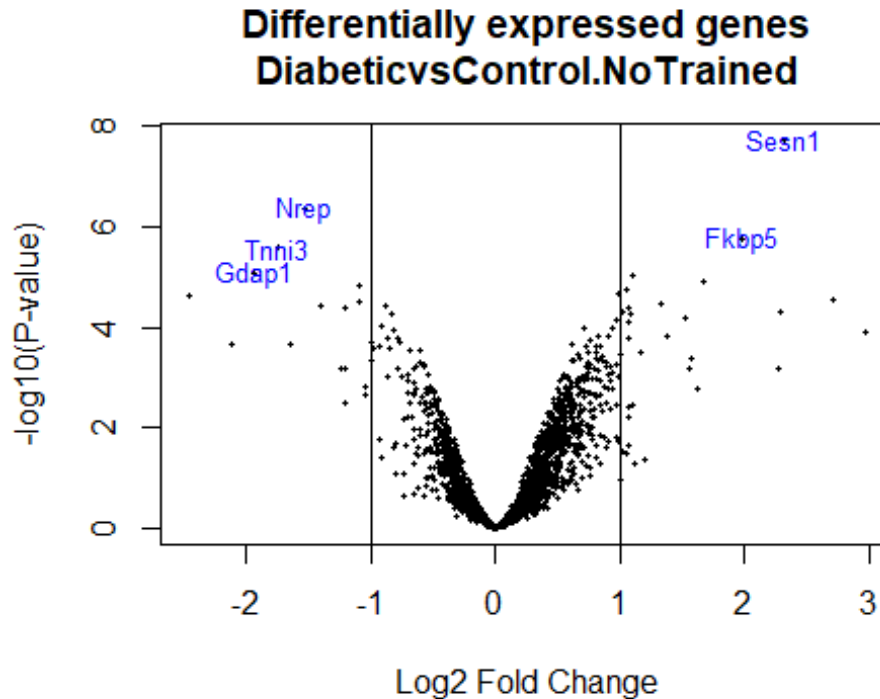
## png
## 2
```

*Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 2:
Ratón diabético vs sano.*

```
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=2, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[2],
sep="\n"))
abline(v=c(-1,1))
```

```
png("Volcano_c2.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=2, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[2],
sep="\n"))
abline(v=c(-1,1))
dev.off()

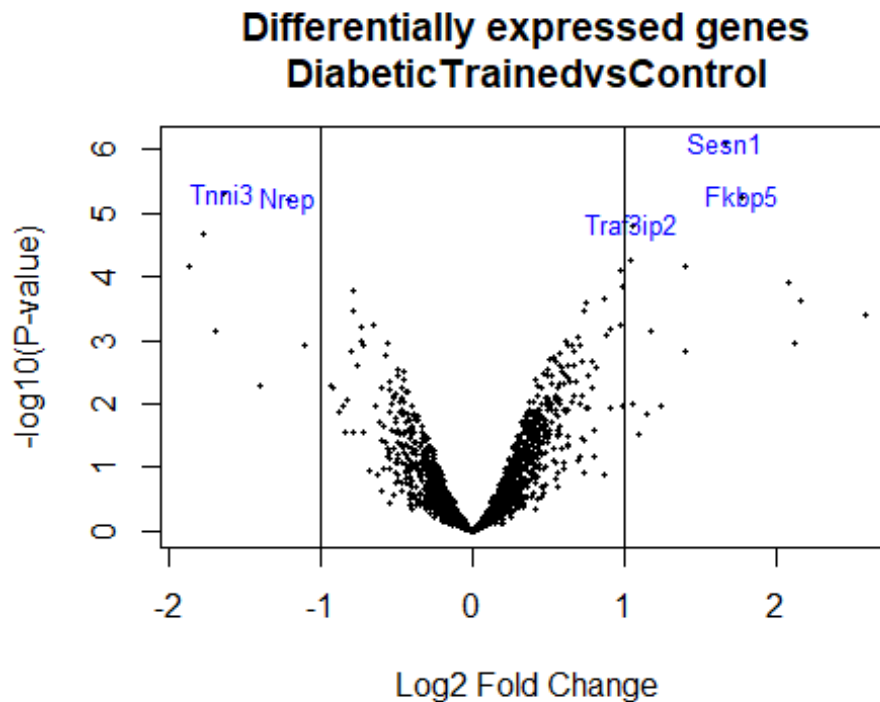
## png
## 2
```

*Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 3:
Ratón diabético con entrenamiento vs ratón sano.*

```
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=3, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[3],
sep="\n"))
abline(v=c(-1,1))
```



```
png("Volcano_c3.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=3, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[3],
sep="\n"))
abline(v=c(-1,1))
dev.off()

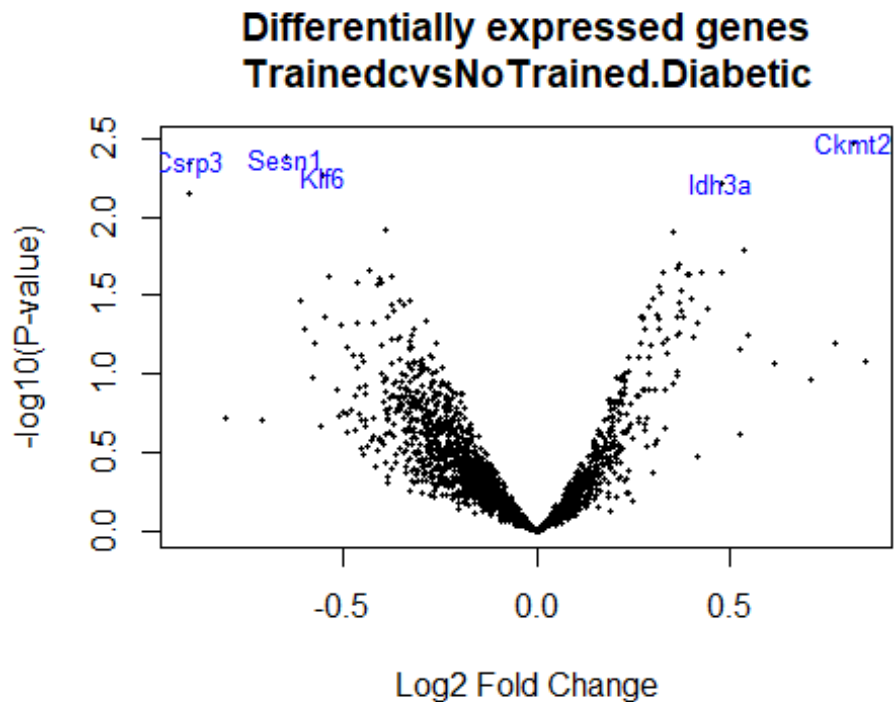
## png
## 2
```

*Volcano plot con los top 5 genes más diferencialmente expresados en la comparación 4:
Ratón diabético con entrenamiento vs ratón diabético sin entrenamiento.*

```
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=4, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[4],
sep="\n"))
abline(v=c(-1,1))
```



```
png("Volcano_c4.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=4, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[4],
sep="\n"))
abline(v=c(-1,1))
dev.off()

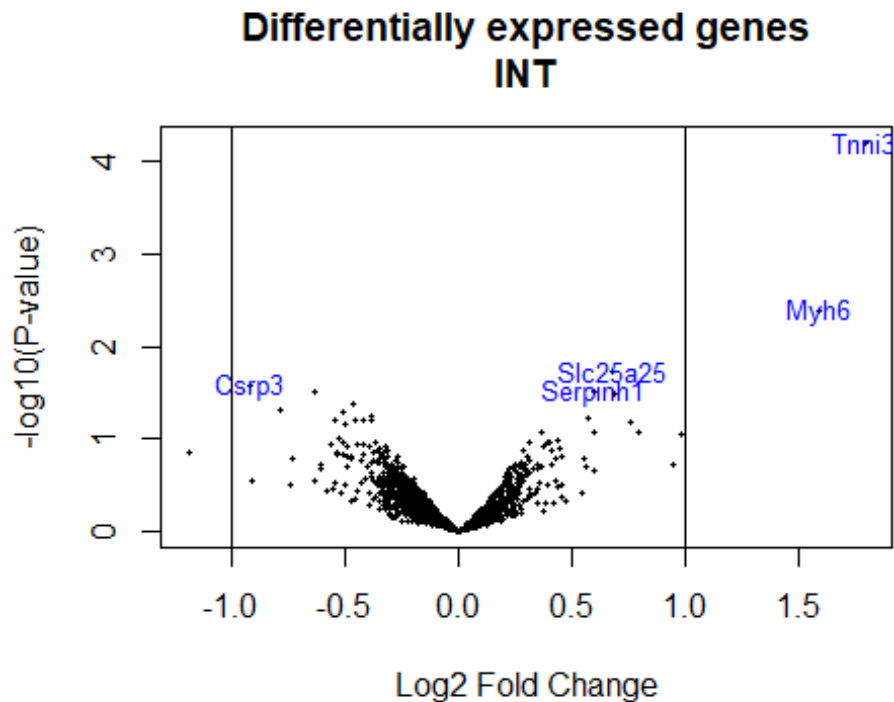
## png
## 2
```

Volcano plot con los top 5 genes más diferencialmente expresados cuando estudiamos la interacción.

```
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=5, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[5],
sep="\n"))
abline(v=c(-1,1))
```



```
png("Volcano_c5.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
geneSymbols <- select(mgu74av2.db, rownames(fit.main), c("SYMBOL"))

## 'select()' returned 1:1 mapping between keys and columns

SYMBOLS<- geneSymbols$SYMBOL
volcanoplot(fit.main, coef=5, highlight=5, names=SYMBOLS,
main=paste("Differentially expressed genes", colnames(cont.matrix)[5],
sep="\n"))
abline(v=c(-1,1))
dev.off()

## png
## 2
```

Comparaciones múltiples y diagrama de Venn

Vamos a estudiar qué genes están up-regulados y down-regulados en cada una de las comparaciones. Estos genes se muestran a continuación:

```
library(limma)
res<-decideTests(fit.main, method="separate", adjust.method="fdr", p.v
alue=0.1, lfc=1)
sum.res.rows<-apply(abs(res),1,sum)
res.selected<-res[sum.res.rows!=0,]
print(summary(res))

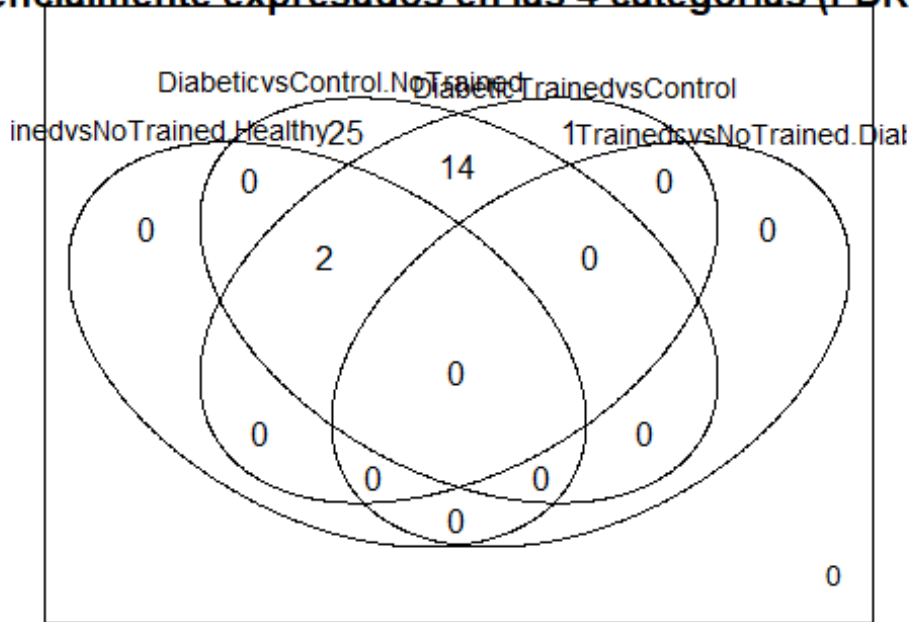
##          TrainedvsNoTrained.Healthy DiabeticvsControl.NoTrained
## Down                                2                             15
## NotSig                             2174                          2135
## Up                                  0                             26
##          DiabeticTrainedvsControl TrainedcvvsNoTrained.Diabetic INT
```

```
## Down          6          0      0
## NotSig       2159       2176  2176
## Up           11          0      0
```

Diagrama de Venn los genes diferencialmente expresados en común entre las 4 categorías con $FDR < 0.1$ y $\log FC > 1$.

```
vennDiagram (res.selected[,1:4], cex=0.90)
title("Genes diferencialmente expresados en las 4 categorías (FDR < 0.1 y logFC > 1)")
```

Genes diferencialmente expresados en las 4 categorías (FDR < 0.1 y logFC > 1)



```
png("VennDiag.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
vennDiagram (res.selected[,1:4], cex=0.90)
title("Genes diferencialmente expresados en las 4 categorías (FDR < 0.1 y logFC > 1)")
dev.off()
```

```
## png
## 2
```

Visualización de los perfiles de expresión usando mapas de calor "Heatmaps"

Heatmap sin ningún tipo de agrupación de muestras. A la derecha podemos ver el listado de los genes que se encuentran up/downregulados.

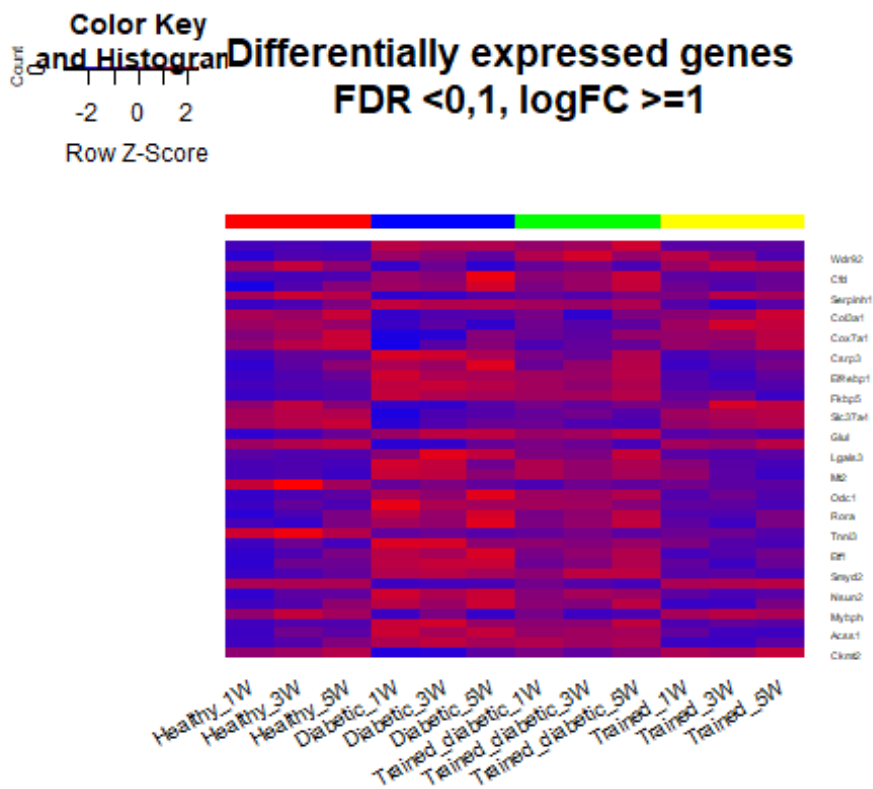
```
probesInHeatmap <- rownames(res.selected)
HMdata <- exprs(eset_filtered)[rownames(exprs(eset_filtered)) %in% probesInHeatmap,]
geneSymbols <- select(mgu74av2.db, rownames(HMdata), c("SYMBOL"))
## 'select()' returned 1:1 mapping between keys and columns
```

```

SYMBOLS<- geneSymbols$SYMBOL
rownames(HMdata) <- SYMBOLS
write.csv(HMdata, file = file.path("./results/data4Heatmap.csv"))

my_palette <- colorRampPalette(c("blue", "red"))(n = 299)
library(gplots)
heatmap.2(HMdata,
  Rowv = FALSE,
  Colv = FALSE,
  main = "Differentially expressed genes \n FDR <0,1, logFC >=1",
  scale = "row",
  col = my_palette,
  sepcolor = "white",
  sepwidth = c(0.05,0.05),
  cexRow = 0.5,
  cexCol = 0.9,
  key = TRUE,
  keysize = 1.5,
  density.info = "histogram",
  ColSideColors = c(rep("red",3),rep("blue",3), rep("green",3), rep("yellow",3)),
  tracecol = NULL,
  dendrogram = "none",
  srtCol = 30)

```



```

png("Heatmap.png", width = 20, height = 12,
  units = "cm", res = 600, pointsize = 10)
heatmap.2(HMdata,
  Rowv = FALSE,
  Colv = FALSE,
  main = "Differentially expressed genes \n FDR <0,1, logFC >=1",

```

```

scale = "row",
col = my_palette,
sepcolor = "white",
sepwidth = c(0.05,0.05),
cexRow = 0.5,
cexCol = 0.9,
key = TRUE,
keysize = 1.5,
density.info = "histogram",
ColSideColors = c(rep("red",3),rep("blue",3), rep("green",3), rep("yellow",3)),
tracecol = NULL,
dendrogram = "none",
srtCol = 30)
dev.off()

## png
## 2

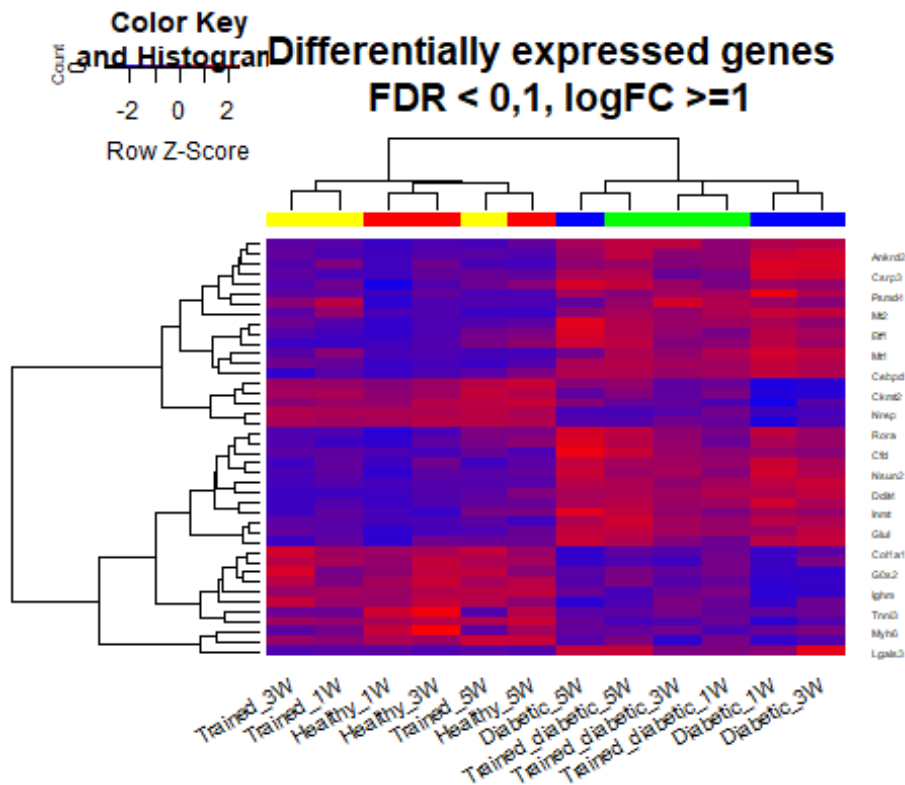
```

Heatmap con dos tipos de agrupamientos: arriba estaría la agrupación entre muestras (columnas) y a la izquierda la agrupación por genes (filas).

```

heatmap.2(HMdata,
Rowv = TRUE,
Colv = TRUE,
dendrogram = "both",
main = "Differentially expressed genes \n FDR < 0,1, logFC >=1",
scale = "row",
col = my_palette,
sepcolor = "white",
sepwidth = c(0.05,0.05),
cexRow = 0.5,
cexCol = 0.9,
key = TRUE,
keysize = 1.5,
density.info = "histogram",
ColSideColors = c(rep("red",3),rep("blue",3), rep("green",3), rep("yellow",3)),
tracecol = NULL,
srtCol = 30)

```



```
png("Heatmap_Dendo.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
heatmap.2(HMdata,
  Rowv = TRUE,
  Colv = TRUE,
  dendrogram = "both",
  main = "Differentially expressed genes \n FDR < 0,1, logFC >=1",
  scale = "row",
  col = my_palette,
  sepcolor = "white",
  sepwidth = c(0.05,0.05),
  cexRow = 0.5,
  cexCol = 0.9,
  key = TRUE,
  keysize = 1.5,
  density.info = "histogram",
  ColSideColors = c(rep("red",3),rep("blue",3), rep("green",3), rep("yellow",3)),
  tracecol = NULL,
  srtCol = 30)
dev.off()

## png
## 2
```

PASO 8: Análisis de significación biológica

```
listOfTables <- list(TrainedvsNoTrained.Healthy = topHealthy.Trainedvs
  Healthy.NoTrained, DiabeticvsControl.NoTrained = topDiabetic.NoTrain
  edvsHealthy.NoTrained, DiabeticTrainedvsControl = topDiabetic.Trainedv
  sHealthy.NoTrained, TrainedcvvsNoTrained.Diabetic = topDiabetic.Trained
  vsDiabetic.NoTrained, INT = topTab_INT)
```



```

listOfSelected <- list()
for (i in 1:length(listOfTables)){
  # select the toptable
  topTab <- listOfTables[[i]]
  # select the genes to be included in the analysis
  whichGenes<-topTab["adj.P.Val"]<0.15
  selectedIDs <- rownames(topTab)[whichGenes]
  # convert the ID to Entrez
  EntrezIDs<- select(mgu74av2.db, selectedIDs, c("ENTREZID"))
  EntrezIDs <- EntrezIDs$ENTREZID
  listOfSelected[[i]] <- EntrezIDs
  names(listOfSelected)[i] <- names(listOfTables)[i]
}

## 'select()' returned 1:1 mapping between keys and columns
## 'select()' returned 1:1 mapping between keys and columns
## 'select()' returned 1:1 mapping between keys and columns
## 'select()' returned 1:1 mapping between keys and columns

sapply(listOfSelected, length)

##   TrainedvsNoTrained.Healthy   DiabeticvsControl.NoTrained
##                               2                             597
##   DiabeticTrainedvsControl   TrainedcvvsNoTrained.Diabetic
##                               81                             0
##                               INT
##                               1

mapped_genes2GO <- mappedkeys(org.Mm.egGO)
mapped_genes2KEGG <- mappedkeys(org.Mm.egPATH)
mapped_genes <- union(mapped_genes2GO , mapped_genes2KEGG)

```

Análisis de significación biológica para las cuatro comparaciones

```

listOfData <- listOfSelected[1:3]
comparisonsNames <- names(listOfData)
universe <- mapped_genes

for (i in 1:3){
  genesIn <- listOfData[[i]]
  comparison <- comparisonsNames[i]
  enrich.result <- enrichPathway(gene = genesIn,
    pvalueCutoff = 0.05,
    readable = T,
    pAdjustMethod = "BH",
    organism = "mouse",
    universe = universe)
  cat("#####")
  cat("\nComparison: ", comparison, "\n")
  print(head(enrich.result))
  if (length(rownames(enrich.result@result)) != 0) {
    write.csv(as.data.frame(enrich.result),
      file =paste0("./results/", "ReactomePA.Results.", comparison, ".csv"),
      row.names = FALSE)
    pdf(file=paste0("./results/", "ReactomePABarplot.", comparison, ".pdf"))
    print(barplot(enrich.result, showCategory = 15, font.size = 4,

```

```

title = paste0("Reactome Pathway Analysis for ", comparison, ". Barplot
"))
dev.off()

pdf(file = paste0("./results/", "ReactomePAcnetplot.", comparison, ".pdf"
))
print(cnetplot(enrich.result, categorySize = "geneNum", schowCategory
= 15,
vertex.label.cex = 0.75))
dev.off()
}
}

## #####
## Comparison: TrainedvsNoTrained.Healthy
##
## ID Description GeneRatio
BgRatio
## R-MMU-390522 R-MMU-390522 Striated Muscle Contraction 2/2
33/8772
## R-MMU-397014 R-MMU-397014 Muscle contraction 2/2 1
77/8772
## R-MMU-5578775 R-MMU-5578775 Ion homeostasis 1/2
52/8772
## R-MMU-5576891 R-MMU-5576891 Cardiac conduction 1/2 1
23/8772
##
## pvalue p.adjust qvalue geneID Count
## R-MMU-390522 1.372512e-05 5.490048e-05 NA Tnni3/Myh6 2
## R-MMU-397014 4.048911e-04 8.097821e-04 NA Tnni3/Myh6 2
## R-MMU-5578775 1.182144e-02 1.576192e-02 NA Tnni3 1
## R-MMU-5576891 2.784874e-02 2.784874e-02 NA Tnni3 1

## #####
## Comparison: DiabeticvsControl.NoTrained
##
## ID
## R-MMU-72613 R-MMU-72613
## R-MMU-72737 R-MMU-72737
## R-MMU-72706 R-MMU-72706
## R-MMU-72689 R-MMU-72689
## R-MMU-156827 R-MMU-156827
## R-MMU-72766 R-MMU-72766
##
Description
## R-MMU-72613 Eukaryotic Translation
Initiation
## R-MMU-72737 Cap-dependent Translation
Initiation
## R-MMU-72706 GTP hydrolysis and joining of the 60S riboso
mal subunit
## R-MMU-72689 Formation of a pool of free 4
0S subunits
## R-MMU-156827 L13a-mediated translational silencing of Ceruloplasmin
expression
## R-MMU-72766 Translation

```

##	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
## R-MMU-72613	28/356	112/8772	3.316585e-15	1.190654e-12	9.303894e-13
## R-MMU-72737	28/356	112/8772	3.316585e-15	1.190654e-12	9.303894e-13
## R-MMU-72706	24/356	105/8772	2.753523e-12	6.590099e-10	5.149572e-10
## R-MMU-72689	22/356	94/8772	1.385929e-11	2.367913e-09	1.850312e-09
## R-MMU-156827	23/356	104/8772	1.648965e-11	2.367913e-09	1.850312e-09
## R-MMU-72766	33/356	217/8772	3.533560e-11	4.228494e-09	3.304189e-09

geneID

R-MMU-72613 Eif4ebp1/Rpl18/Rps23/Rps4x/Eif4b/Rpl6/Rpl18/Eif2b1/Rps28/Rps19/Eif3c/Rplp0/Rps18/Eif2b5/Eif3g/Eif5b/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1/Eif2b2

R-MMU-72737 Eif4ebp1/Rpl18/Rps23/Rps4x/Eif4b/Rpl6/Rpl18/Eif2b1/Rps28/Rps19/Eif3c/Rplp0/Rps18/Eif2b5/Eif3g/Eif5b/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1/Eif2b2

R-MMU-72706 Rpl8/Rps23/Rps4x/Eif4b/Rpl6/Rpl18/Rps28/Rps19/Eif3c/Rplp0/Rps18/Eif3g/Eif5b/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1

R-MMU-72689 Rpl8/Rps23/Rps4x/Rpl6/Rpl18/Rps28/Rps19/Eif3c/Rplp0/Rps18/Eif3g/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1

R-MMU-156827 Rpl8/Rps23/Rps4x/Eif4b/Rpl6/Rpl18/Rps28/Rps19/Eif3c/Rplp0/Rps18/Eif3g/Eif3l/Rps5/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1

R-MMU-72766 Eif4ebp1/Rpl18/Etf1/Eef2/Rps23/Rps4x/Eif4b/Rpl6/Rpl18/Eef1a1/Eif2b1/Srp68/Rps28/Rps19/Eif3c/Rplp0/Rps18/Eif2b5/Eif3g/Eif5b/Eif3l/Rps5/Mrp145/Rpl19/Rps11/Rps3/Eif3b/Rpl7/Rps7/Rpl5/Rps10/Rps3a1/Eif2b2

##	Count
## R-MMU-72613	28
## R-MMU-72737	28
## R-MMU-72706	24
## R-MMU-72689	22
## R-MMU-156827	23
## R-MMU-72766	33

#####

Comparison: DiabeticTrainedvsControl

ID

R-MMU-1474290 R-MMU-1474290

R-MMU-2022090 R-MMU-2022090

R-MMU-1650814 R-MMU-1650814

R-MMU-8948216 R-MMU-8948216

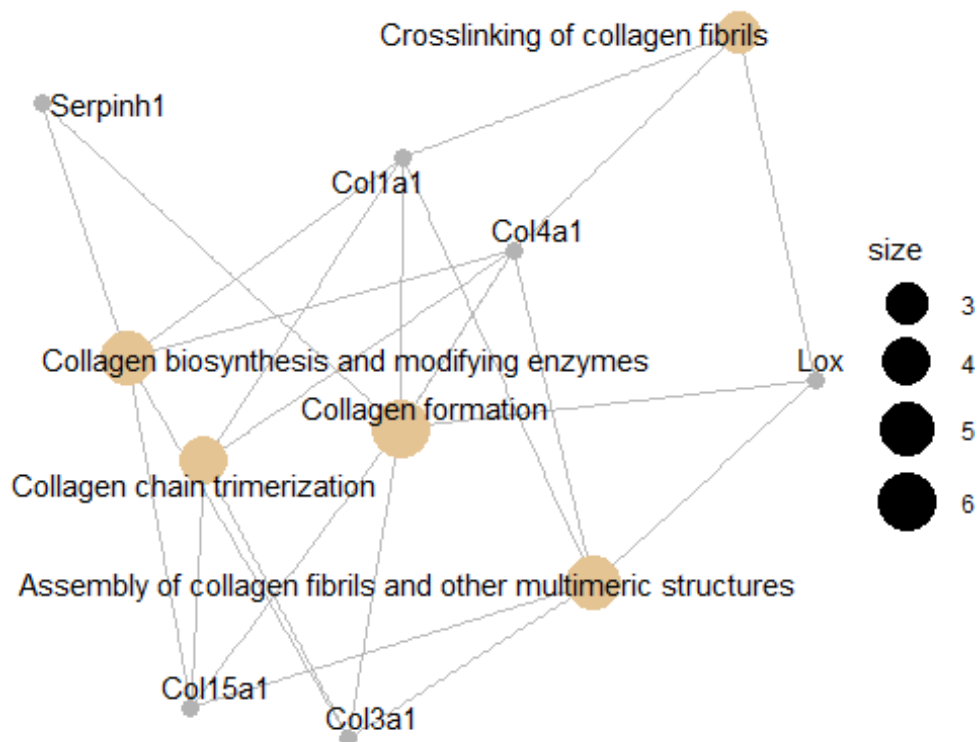
R-MMU-2243919 R-MMU-2243919

R-MMU-1442490 R-MMU-1442490

##						Desc
ription						
## R-MMU-1474290						Collagen fo
rmation						
## R-MMU-2022090	Assembly of collagen fibrils and other multimeric str					
uctures						
## R-MMU-1650814						Collagen biosynthesis and modifying
enzymes						
## R-MMU-8948216						Collagen chain trimer
ization						
## R-MMU-2243919						Crosslinking of collagen
fibrils						
## R-MMU-1442490						Collagen degr
adation						
##	GeneRatio	BgRatio	pvalue	p.adjust	qvalu	
e						
## R-MMU-1474290	6/53	81/8772	8.349036e-06	0.002529758	0.00224984	
6						
## R-MMU-2022090	5/53	57/8772	2.191559e-05	0.002622532	0.00233235	
4						
## R-MMU-1650814	5/53	59/8772	2.596566e-05	0.002622532	0.00233235	
4						
## R-MMU-8948216	4/53	39/8772	8.353404e-05	0.006327703	0.00562755	
6						
## R-MMU-2243919	3/53	18/8772	1.594089e-04	0.009660180	0.00859130	
1						
## R-MMU-1442490	4/53	55/8772	3.225077e-04	0.016286640	0.01448455	
8						
##				geneID	Count	
## R-MMU-1474290	Col15a1/Serpinh1/Col1a1/Lox/Col4a1/Col3a1				6	
## R-MMU-2022090	Col15a1/Col1a1/Lox/Col4a1/Col3a1				5	
## R-MMU-1650814	Col15a1/Serpinh1/Col1a1/Col4a1/Col3a1				5	
## R-MMU-8948216	Col15a1/Col1a1/Col4a1/Col3a1				4	
## R-MMU-2243919	Col1a1/Lox/Col4a1				3	
## R-MMU-1442490	Col15a1/Col1a1/Col4a1/Col3a1				4	

Red de interacción génica

```
cnetplot(enrich.result, categorySize = "geneNum", schowCategory = 15,
vertex.label.cex = 0.75)
```



```
png("Enrichr_red.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
cnetplot(enrich.result, categorySize = "geneNum", schowCategory = 15,
vertex.label.cex = 0.75)
dev.off()
```

```
## png
## 2
```